TOSCA

Version 5.0

Installation Guide

FE-DESIGN GmbH
http://www.fe-design.de

1 CD content 7

2 General 8

2.1 Update TOSCA 5.0 8
2.2 Supported products 8
2.3 Supported platforms 10

3 System Requirements 11

3.1 Operating systems and runtime libraries 11
3.2 Java 12

4 Preparing the installation 14

4.1 UNIX Systems 15
4.1.1 DISPLAY Variable 15
4.1.2 IATEMPDIR Variable 15
4.1.3 Mount the CD-ROM 15
4.1.4 Selection of the installer 16
4.2 Windows Systems 16

5 Procedures for TOSCA installation 17

5.1 Selection of the components to install 18
5.2 TOSCA Installation 20
5.2.1 Selection of the installation directory 20
5.2.2 Selection of the platform 20
5.2.3 Selection of the solver interfaces 23
5.2.4 Selection of the default solver 25
5.2.5 Configuration of the solver calls 25
5.2.5.1 ABAQUS 26
5.2.5.2 MSC.Nastan 27
5.2.5.3 ANSYS 28
5.2.5.4 I-DEAS 29
5.2.5.5 FALANCS 31
5.2.5.6 FEMFAT 32
5.2.6 Selection of the additional TOSCA modules 34
5.2.6.1 TOSCA.gui 34
5.2.6.2 TOSCA.post 34
5.2.6.3 TOSCA.client 35

5.2.7 Selection of the Web Browser 35
5.2.8 Selection of the Acrobat Reader 35
5.2.9 Generation of TOSCA links 35
5.3 TOSCA Personal Installation 37
5.3.1 Selection of the installation directory 37
5.3.2 Selection of the solver interfaces 38
5.3.3 Configuration of the solver calls 39
5.3.3.1 Ansys 39
5.3.3.2 MSC.visual Nastran for Windows (vN4W) 40
5.3.3.3 I-DEAS 44
5.3.4 Selection of the WEB BROWSER 47
5.3.5 Selection of the Acrobat Reader 47
5.3.6 Generation of the TOSCA links 47
5.4 Installation of the FlexLM license server 48
5.4.1 Automatic installation of the FlexLM server 50
5.4.2 Required structure of the boot-scripts on the UNIX target system 50
5.4.3 Installation on Windows systems 51
5.5 Summary 51

6 Licensing 52

6.1 FlexLM in general 52
6.2 Preparation 52
6.3 Determination of the FlexLM hostid 52
6.4 FlexLM Software 53
6.5 TOSCA tools 53
6.6 Installation of a “node-lock“ license 54
6.7 Installation of a license server 54
6.8 Installation of redundant license servers 56
6.9 Configuration of TOSCA 56
6.10 Test of the installation 57
6.11 Note 57

7 Test of the installation 57

Version 5.0 Rev. 01 - 03.2004

8 Appendix 62
8.1 File structure in the installation directory 62
8.1.1 Directories and files in the TOSCA installation folder 62
8.1.2 Files in the $tosca/bin directory 68
8.2 Configuration files 71
8.2.1 tosca_ctrl.cfg 71
8.2.2 tosca_post.cfg 80
8.2.3 Startscripts 81
8.2.3.1 Unix 81
8.2.3.2 Windows 83
8.3 Troubleshooting 83
8.3.1 Configuration 83
8.3.1.1 tosca50 or other tosca modules cannot be found 83
8.3.1.2 Finite element solver executables not defined correctly 84
8.3.1.3 Path to runtime libraries is not correct 84
8.3.1.4 /tmp directory is full 85
8.3.1.5 java virtual machines are not accessible 85
8.3.2 Licensing 86
8.3.2.1 Checklist 87
8.3.2.2 TOSCA Test utilities 88
8.3.2.3 Flexlm Test utilities 88

Version 5.0 Rev. 01 - 03.2004

1 CD content

The following directories may be found on the CD:

- Info:
  - Installation Guide English
  - Installation Guide German
- Disk1/InstData
  - AIX/VM

    `install.bin installation program for AIX 4.3.3`
  - HPUX/VM

    `install.bin installation program for HPUX 11 PA`
  - IRIX/VM

    `install.bin installation program for Irix 6.5`
  - Linux/VM

    `install.bin installation program for Linux 32-Bit`
  - Solaris/VM

    `install.bin installation program for Solaris 8/9`
  - Windows/VM

    `install.exe installation program for Windows 2000/XP`
- Flexlm

  `flexlm_docu.tar.gz  Flexlm user documentation`

  `flexlm_aix.tar.gz Flexlm files or AIX`

  `flexlm_hpux.tar.gz Flexlm files for HP-UX`

  `flexlm_irix.tar.gz Flexlm files for Irix`

  `flexlm_linux.tar.gz Flexlm files for Linux`

  `flexlm_solaris.tar.gz Flexlm files for Solaris`

  `flexlm_winnt.tar.gz Flexlm files for Windows`

LIESMICH.txt

README.txt

Version 5.0 Rev. 01 - 03.2004

2 General

The TOSCA 5.0 Installation is based on InstallAnywhere (ZeroG). The online documentation is available in pdf-format. The CD may be accessed locally or via NFS mount.

TOSCA Version 5.0 uses the FlexLM license management system from Macrovision. General information regarding the installation and operation of the Flex LM license manager can be found in the 'enduser.pdf' document on the CD.

In addition to the installation program, all necessary components exist in a complete tar archive for the platforms supported. These archive files may be copied to the target platform and extracted manually. Further information may be found in this installation document. Information about the FlexLM License Management can be found in the FlexLM nuser manual on the cd.

2.1 Update TOSCA 5.0

If you are already using TOSCA Version 4.5, then you are already familiar with the installation procedure. The fastest way to update to TOSCA 5.0 is as follows:

- First determine the FlexLM license server and submit the FlexLM host ID to FE-Design. We will send you the 'license.dat' file which contains the actualized license for TOSCA 5.0.
- The license server may be set up manually or by using the installation program.
- The procedures for installation are very similar to that of TOSCA 4.5. All information regarding the solvers and the pre and postprocessing tools are available.
- After completion of the TOSCA installation, a test may be run by loading tosca50_gui (graphical user interface) or tosca50 -help (commandline)

2.2 Supported products

FE solver:

- Abaqus
- Ansys
- I-deas (SAS Solver)
- Marc
- Nastran
- Permas

Other simulation software

- Falancs
- Femfat

Pre / Postprocessing

- Abaqus CAE
- Ansys
- I-deas
- Medina
- MSC.Patran
- MSC.visual Nastran for Windows
- Femap

Note:

The products mentioned above have to be completely installed and configured when used in combination with TOSCA.

Abaqus:

ABAQUS does not automatically delete existing result files per standard. To avoid this, ABAQUS asks interactively if result files can be deleted during each step of the optimization procedure. The following setting has to be added to the abaqus_v6.env configuration file:

```
ask_delete=OFF
```

I-Deas:

The installation and configuration of the I-DEAS Stand Alone Solver (SAS) can be found in the I-DEAS Installation Guide

MSC.Nastran on Windows systems:

If TOSCA runs on a Windows system and MSC.Nastran is used as a finite element solver, the following Nastran executable file must be used:

```
..../msc2001/i386/nastran.exe
```

Ansys on Windows systems:

If TOSCA runs on a Windows system and ANSYS is used as a finite element solver, the following Ansys executable file must be used:

```
ansys57.exe, bzw. 60,61,70, 71, etc.
```

Version 5.0 Rev. 01 - 03.2004

2.3 Supported platforms

Tosca 5.0 is available on the following platforms:

- Windows 2000
- Windows XP
- HPUX 11 PA Risc
- IRIX 6.5.16m MIPS
- AIX 4.3.3 POWER
- Solaris 8.0 SPARC
- Linux (> 2.4.10) Intel 32 Bit

At the beginning of the installation, all external products that are to be used in combination with TOSCA should be correctly installed and configured on the target platform. If TOSCA is to be installed for multiple platforms, TOSCA assumes that the external products all have the same program calls (program name and program path)

3 System Requirements

TOSCA 5.0 is based on different technologies. For single components, enhanced system requirements have to be fulfilled.

3.1 Operating systems and runtime libraries

The core modules of TOSCA consist of native C/C++ and Fortran77/Fortran90 executable programs which have the following system requirements.

HPUX 11 :

Base is the General Release Patch 1999, XSWGR1100 for which

the following compilers were used:

- HP C/ANSI C ... B.11.01.20
- FORTRAN ... B.11.01.20
- FORTRAN 90... B.11.01.20
- HP aC++ ... C.03.25

If a problem with the runtime environment occurs, the installed versions on your platform may be detected as follows:

```
what /usr/lib/libCsup.2
```

The version must be >= A.03.26

and for use of the ONF interface programs (all solvers except for MSC.Nastran)

```
# what /usr/lib/pa20_64/libcl.2
/usr/lib/pa20_64/libcl.2:
        libIO77 HP HPUX [ Release B.11.01.12 PA 64bit ]
         (hp700:hp/ux) Mar  1 2002
         Copyright (c) 2001 Hewlett Packard.
        libcl.sl version B.11.01.15 - Feb 27, 2002
         Trap Library version UX.11.01.05 - 01/10/23
         Unwind Library version UX.11.01.05 - 00/08/15
```

FE Design uses the patch : PHCO_24148, libc cumulative patch

IRIX 6.5 :

The IRIX version is based on Irix 6.5.16m

and was created with the MipsPro Compiler Version 7.3.1.3m.

The following packages must be installed on your Irix compute nodes.

Standard Execution Environment (C++ Headers and Libraries 7.3.1.3m) N32 and 64bit

Standard Execution Environment (Fortran Headers and Libraries 7.3.1.3m)

AIX 4.3.3

The AIX version is based on Aix 4.3.3

and was created with the following compilers

XLC 5.0.2.2 C-Compiler

VACPP 5.0.2.0 C++ - Compiler

XLF 7.1.1.0 Fortran77/90 Compiler

The necessary runtime libraries are delivered with TOSCA and the library paths are extended in the start scripts (LD_LIBPATH)

Solaris

The SOLARIS Version was created using Solaris 8.0

The following compilers were used:

Forte 6 update 2 C/C++/F77/F90

The necessary runtime libraries are delivered with TOSCA and the library paths are extended in the startscripts (LD_LIBRARY_PATH)

Linux

the LINUX Version is based on Suse 7.3 Kernel 2.4.10

The following compilers were used:

Intel C/C++ Version 7.1

Intel Fortran Compiler Version 7.1

The necessary runtime libraries are delivered with TOSCA and the library paths are extended in the startscripts (LD_LIBRARY_PATH)

Windows

TOSCA was created on Windows 2000 and Windows XP

3.2 Java

Java is necessary for the installation of and for use of several of the TOSCA modules:

All installation programs contain self-extracting Java virtual machines. The installation starts with the extraction of the appropriate VM in a temporary directory, either in the system default directory or a directory defined with the IATEMPDIR environment variable. This VM is only used to run the installation.

The components of the TOSCA graphical user interface require Java virtual machines for each client platform. TOSCA installation contains Java VM's for each supported platform. These are installed in the TOSCA directory and are only visible for TOSCA components. Your system will not be changed and other Java-based applications will not be influenced. The installation of Java VM's in the TOSCA directory is recommended because TOSCA is tested with the delivered VM's and you do not need to install Java on each client computer.

Version 5.0 Rev. 01 - 03.2004

4 Preparing the installation

Installation is carried out using self-extracting archives that contain all the necessary components for TOSCA. You can open the appropriate installer either with a file manager (Windows Explorer or Unix equivalent) or call it up on the command line using a console.

TOSCA makes use of several third party products. To simplify the installation it is recommended to gather the information about the installation paths of the products you want to use with TOSCA.

Example: Configuration at FE Design

product	Windows	Unix
Abaqus	C:\Programme\abaqus\Commands\abaqus.bat	/app/abaqus/Commands/abaqus
Nastran	C:\Programme\msc\msc2004\i386\nastran.exe	/usr/bin/nastran
Ansys	C:\Programme\Ansys Inc\ANSYS80\bin\intel\ansys80.exe	/app/ansys_inc/v71/ansys/bin/ansys71
I-Deas	C:\SDRC\I-DEAS9	/app/sdrc/ideas9
Permas	C:\Programme\Intes\PERMAS\procs\permas.bat	/app/intes/permas/procs/permas
Marc	C:\Programme\Msc\marc2003\bin\marc.exe	/app/msc/marc/marc2003/tools/run_marc
Femfat	C:\Programme\ECS\FEMFAT4.4e	/app/femfat/femfat44e
Falancs	C:\Programme\LMS	/app/lms/2.11
Web Browser	C:\Programme\Internet Explorer\IEXPLORE.EXE	/usr/bin/netscape
Acrobat Reader	C:\Programme\Adobe\Acrobat 5.0\Reader\AcroRd32.exe	/usr/bin/acroread
Patran	C:\Programme\msc\patran2001r3\bin\patran.exe	/usr/bin/patran
License Server	6666@fed-ex2	6666@fed-ex2

4.1 UNIX Systems

4.1.1 DISPLAY Variable

As the installation is based on X11, it must be possible to open a X-Window on your terminal. The settings are correct if the <xterm> command opens a xterm window on your local display, otherwise the DISPLAY variable has to be set as follows:

```
DISPLAY=192.168.0.31:0.0
export DISPLAY
```

4.1.2 IATEMPDIR Variable

The installation requires up to 250MB temporary disk space. By default the /tmp directory or the root / is used. If there is insufficient disk space, the temporary directory can be defined using the IATEMPDIR environment variable.

```
IATEMPDIR=/scratch
export IATEMPDIR
```

\+ At the completion of installation, some temporary files may remain in the temporary directory. These files should be deleted manually.

4.1.3 Mount the CD-ROM

\+ The TOSCA CD was generated with so-called Rockridge extensions. The computer used for installation must support them. Correct loading of the CD will be shown in the root file on the installation CD. The path names must appear case-sensitive and must not contain any version markers (;).

HP-UX

```
mount -rF cdfs /dev/... /CDROM
```

IRIX

automatic mount

AIX

mount -v cdrfs -r `/dev/...` /CDROM

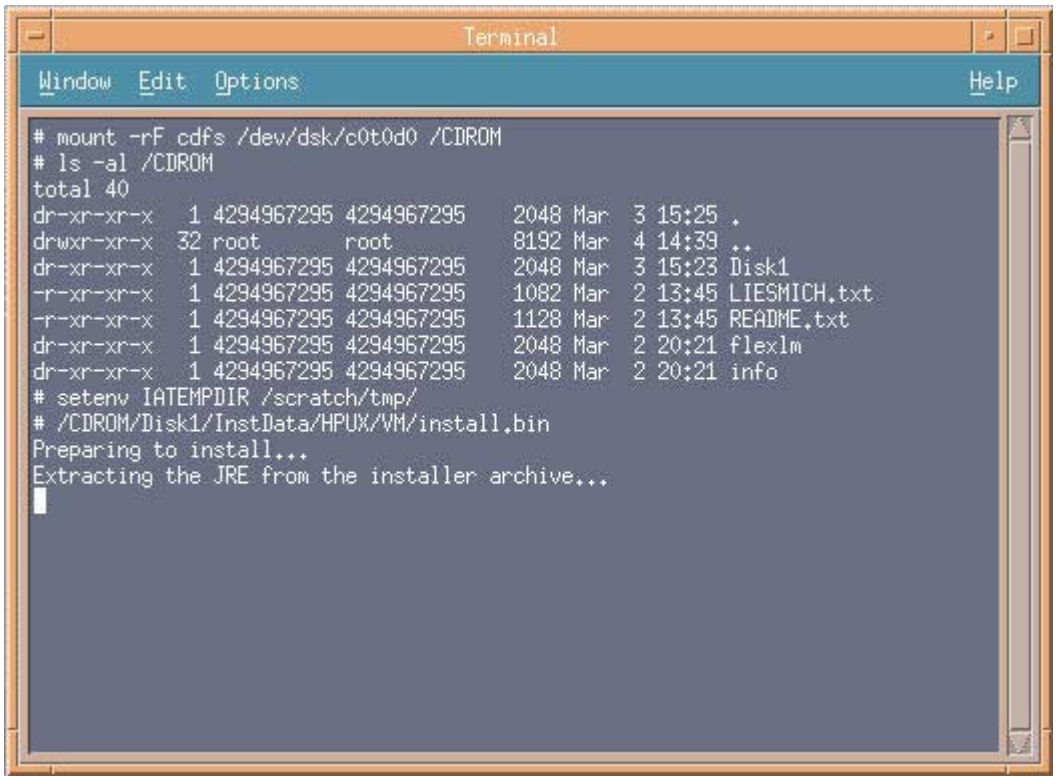
Solaris

mount -F hsfs -r `/dev/...` /CDROM

Linux

mount -t iso9660 `/dev/...` /CDROM

+ Example HPUX:


Terminal
Window Edit Options Help
mount -rF cdfs /dev/dsk/c0t0d0 /CDROM
ls -al /CDROM
total 40
dr-xr-xr-x 1 4294967295 4294967295 2048 Mar 3 15:25 .
drwxr-xr-x 32 root root 8192 Mar 4 14:39 ..
dr-xr-xr-x 1 4294967295 4294967295 2048 Mar 3 15:23 Disk1
-r-xr-xr-x 1 4294967295 4294967295 1082 Mar 2 13:45 LIESMICH.txt
-r-xr-xr-x 1 4294967295 4294967295 1128 Mar 2 13:45 README.txt
dr-xr-xr-x 1 4294967295 4294967295 2048 Mar 2 20:21 flexlm
dr-xr-xr-x 1 4294967295 4294967295 2048 Mar 2 20:21 info
setenv IATEMPDIR /scratch/tmp/
/CDROM/Disk1/InstData/HPUX/VM/install.bin
Preparing to install...
Extracting the JRE from the installer archive...

4.1.4 Selection of the installer

The installers may be found in the platform directories:

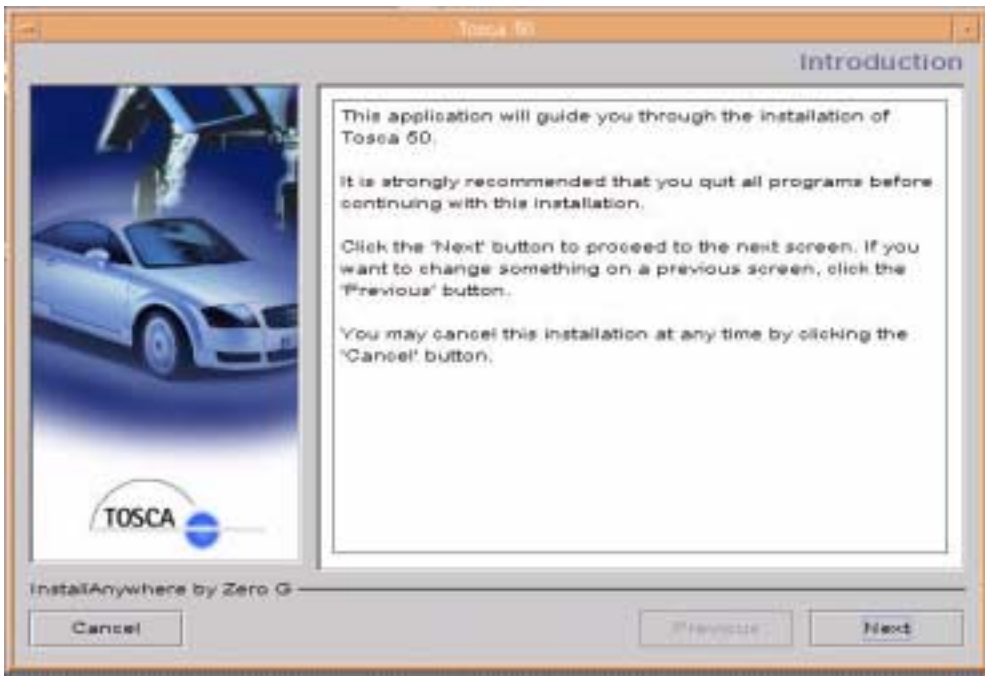
4.2 Windows Systems

No further preparation is necessary for Windows systems.

5 Procedures for TOSCA installation

The installation of TOSCA 5.0 is demonstrated on a HP-UX workstation shown in the example below.

+ Installation is based on the program Java. The directory selection dialogs may show unusual results during installation. The directories have to be marked in the dialog window. When the directory name appears in the ’file name’ text box, it has to be selected using the SELECT button. It is not possible to select directories with a double mouse click.
The installation launches a start window. By clicking the button ’NEXT’, the procedures for TOSCA 5.0 installation are initiated..


Introduction
This application will guide you through the installation of Tosca 60.
It is strongly recommended that you quit all programs before continuing with this installation.
Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.
You may cancel this installation at any time by clicking the 'Cancel' button.
TOSCA
InstallAnywhere by Zero G
Cancel
Previous
Next

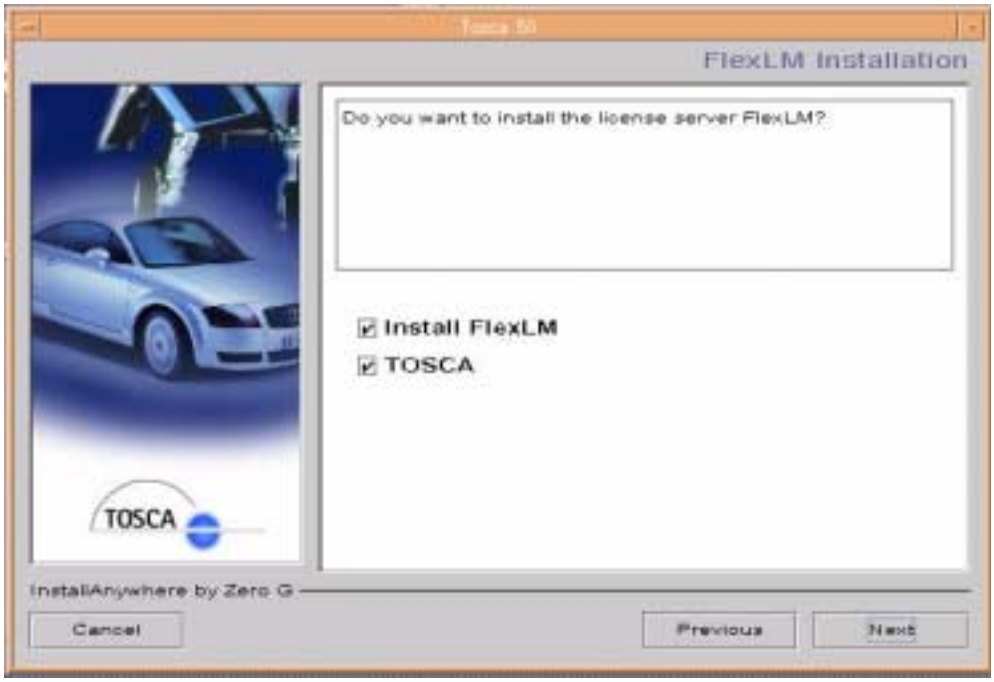
5.1 Selection of the components to install

The component to install has to be selected at the start of installation. The following components may be installed:

Two TOSCA versions are available:

- TOSCA is the full version for all supported platforms and all supported interfaces. The procedures for installation are described in detail in Kapitel 5.2.
- TOSCA Personal is a version of TOSCA which can only be used for Windows Systems in combination with MSC.visual Nastran for Windows, Ansys and I-Deas. A detailed description of the particular features pertaining to the installation of the TOSCA Personal Edition can be found in Kapitel 5.3.

Version 5.0 Rev. 01 - 03.2004

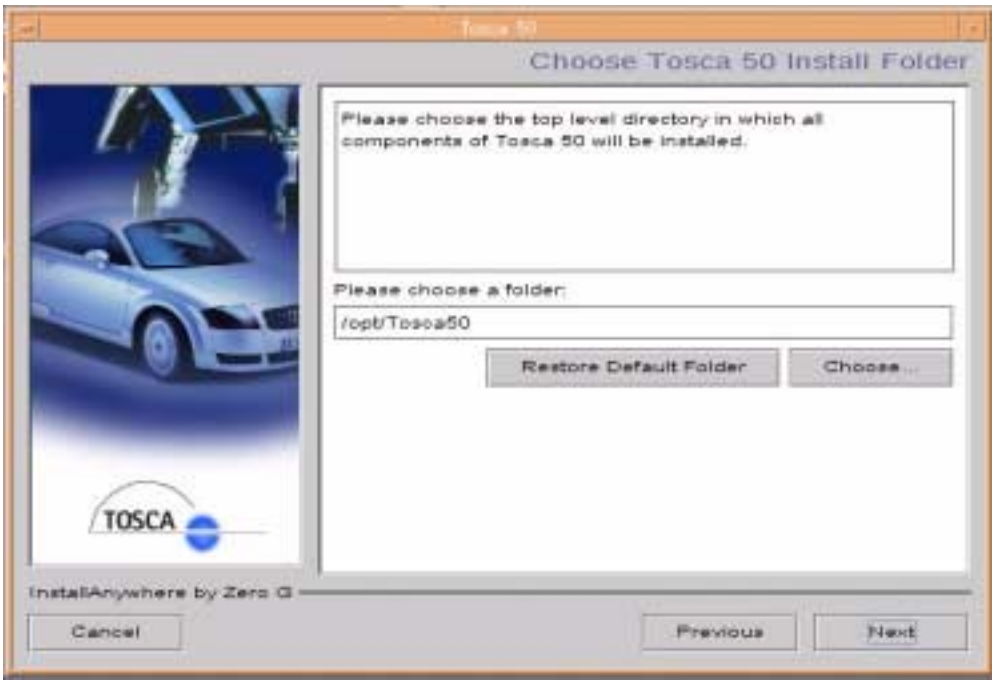


- Install FlexLM
  This installs the components that are necessary to execute the Flex LM license server on the platform where the installation is executed. For example, if the installation is executed on a HP-UX system, the Flex LM components for HP-UX will be copied into the TOSCA installation directory. Installing the Flex LM license manager may also be performed independent to the regular TOSCA installation procedure. At this point of installation, only the necessary components are copied and the flex server is not started. The Flex LM license server may be started at the end of the TOSCA installation.
- TOSCA
  Installation of the TOSCA 5.0 core applications for supported platforms. The solver and postprocessing interfaces have to be selected in the following dialogs.

5.2 TOSCA Installation

5.2.1 Selection of the installation directory

All TOSCA files will be extracted to the selected directory which must be accessable on the computer where the installation is executed.

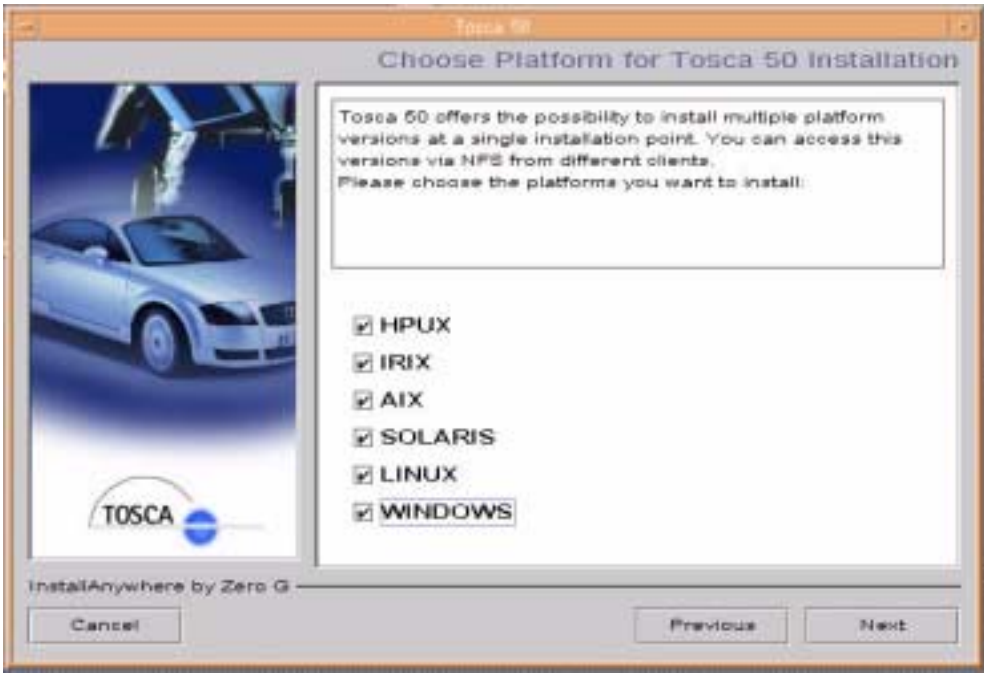


5.2.2 Selection of the platform

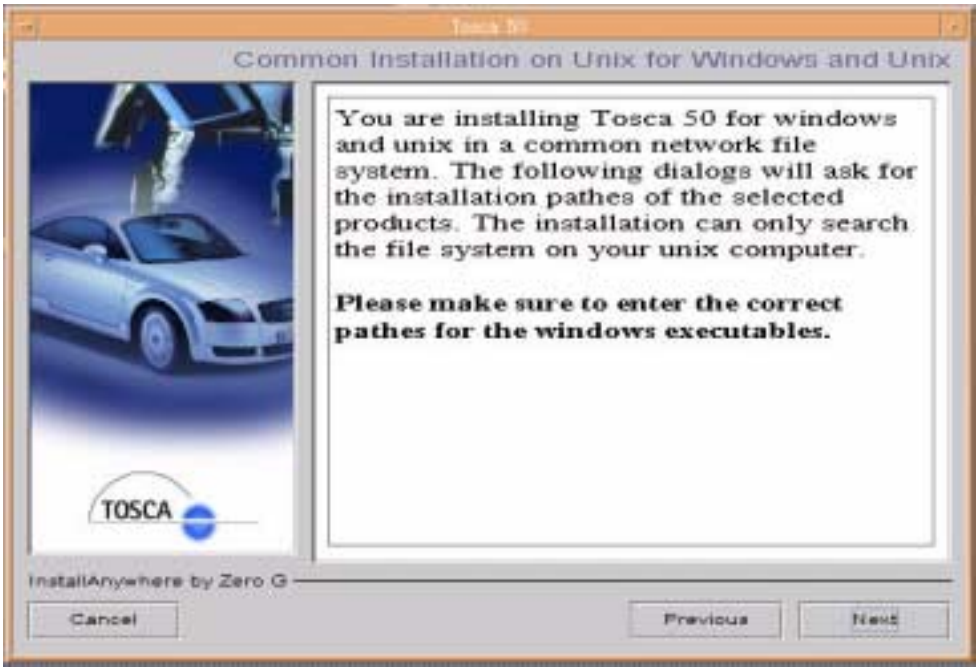
TOSCA may be installed at a single point of installation for multiple platforms in a heterogeneous network. The TOSCA installation directory has to be mounted via NFS on all workstations which must have access to TOSCA.

The platforms which should be supported can be selected in this dialog. If installation is made in a common directory for UNIX and WINDOWS systems, FE-Design recommends that installation be made on the UNIX system and the corresponding directory be exported using Samba or Unix Services for Windows. Another limitation on common installation procedures involves the different path names used in Unix and Windows. This means that the path settings for Unix AND Windows have to be configured. In cases of installation in a common directory, the

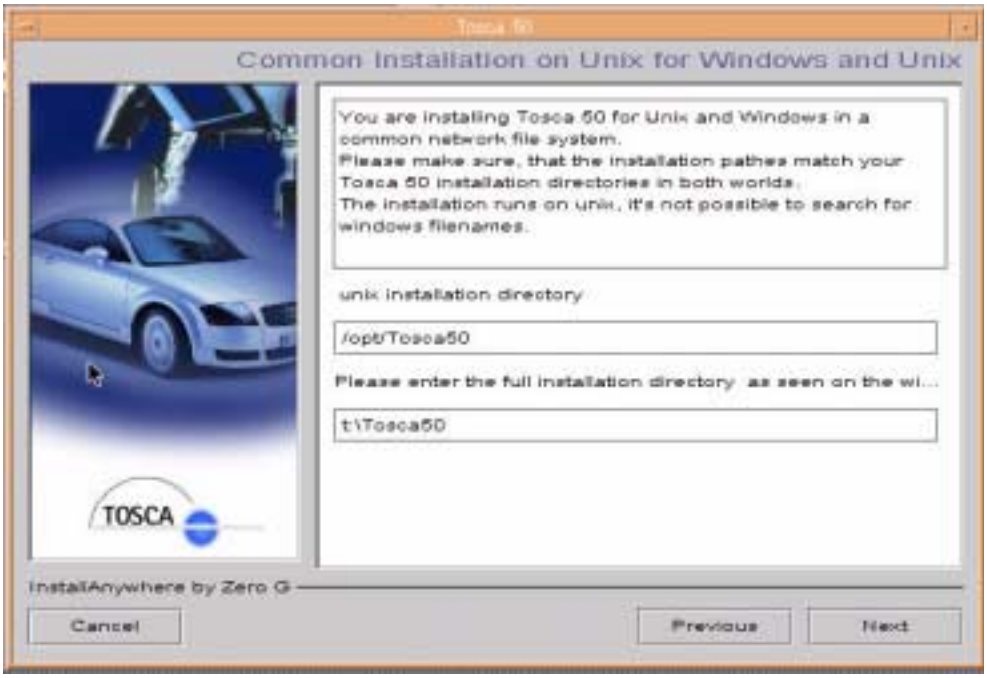
installer first asks for the Windows path of the components then for the Unix path of the components.



The TOSCA components will be copied into the platform specific subdirectories os_<platform>


Common Installation on Unix for Windows and Unix
You are installing Tosca 50 for windows and unix in a common network file system. The following dialogs will ask for the installation pathes of the selected products. The installation can only search the file system on your unix computer.
Please make sure to enter the correct pathes for the windows executables.
TOSCA
InstallAnywhere by Zero G
Cancel
Previous
Next

If the TOSCA application is installed for Windows and Unix in a common directory, great care must be taken in carrying out the following steps. For external products, the installer will first ask for the paths on the Windows system and then for the paths on the Unix system.


Common Installation on Unix for Windows and Unix
You are installing Tosca 50 for Unix and Windows in a common network file system.
Please make sure, that the installation pathes match your Tosca 50 installation directories in both worlds.
The installation runs on unix, it's not possible to search for windows filenames.
unix installation directory
/opt/Tosca50
Please enter the full installation directory as seen on the wi...
t:\Tosca50
TOSCA
InstallAnywhere by Zero G
Cancel
Previous
Next

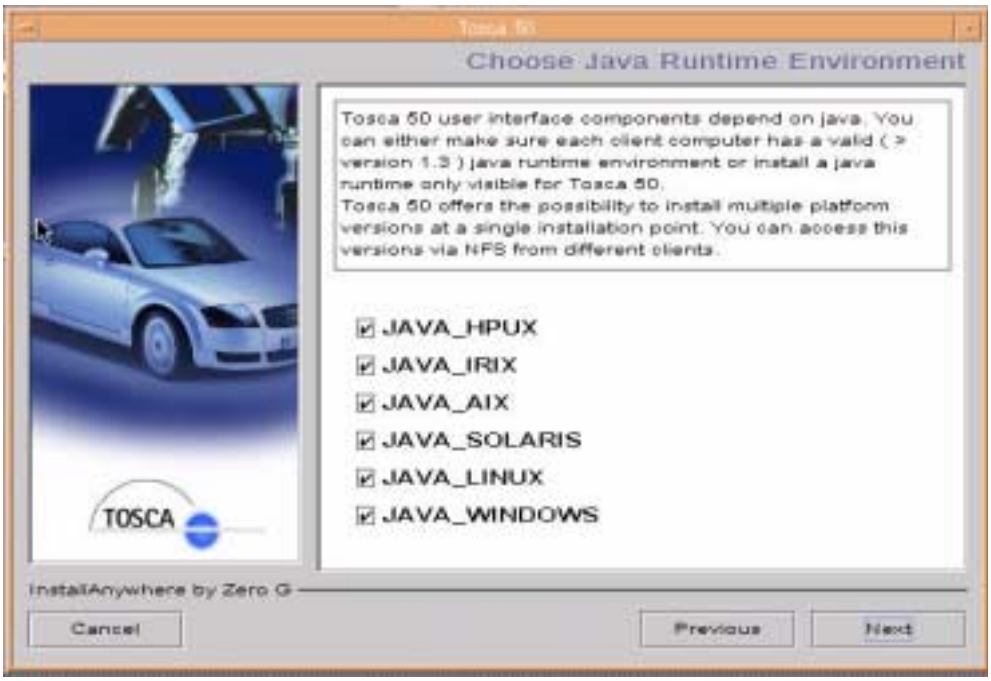
Due to differing path specifications on Windows and Unix systems, paths must be defined for both platforms. The path settings will be stored in two different files:

For the TOSCA core application, the environment variable $tosca is defined in the files:

Unix: <tosca_install_dir>/bin/tosca5_script

Windows: <tosca_install_dir>\bin\common_env.cmd

For graphical applications, the path is stored in the “*.lax“ files in the <tosca_install_dir>/bin/os_<platform> directories.

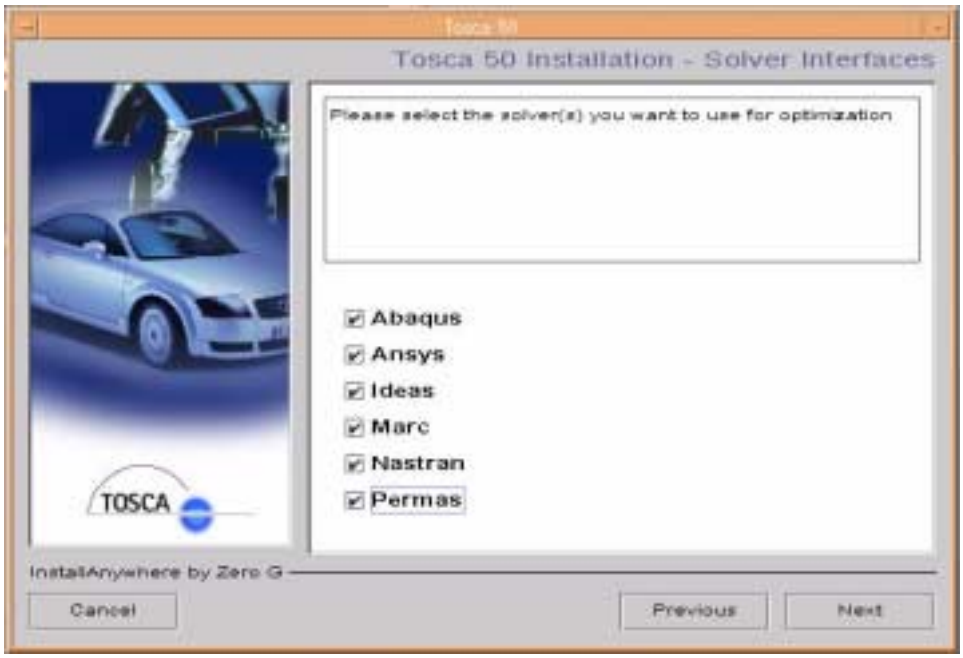

Choose Java Runtime Environment
Tosca 50 user interface components depend on java. You can either make sure each client computer has a valid (> version 1.3) java runtime environment or install a java runtime only visible for Tosca 50.
Tosca 50 offers the possibility to install multiple platform versions at a single installation point. You can access this versions via NFS from different clients.
JAVA_HPUX
JAVA_IRIX
JAVA_AIX
JAVA_SOLARIS
JAVA_LINUX
JAVA_WINDOWS
TOSCA
InstallAnywhere by Zero G
Cancel
Previous
Next

This dialog offers you the possibility of installing Java virtual machines for the platforms requested. If the Java VM is installed on a shared network directory there is no need to install JVM's on each of the client computers. Selection of platforms in this dialog is independent from the platform the setup runs on. This Java installation is only visible for TOSCA-user interfaces, it doesn't change your system and has no impact on other Java-based applications.

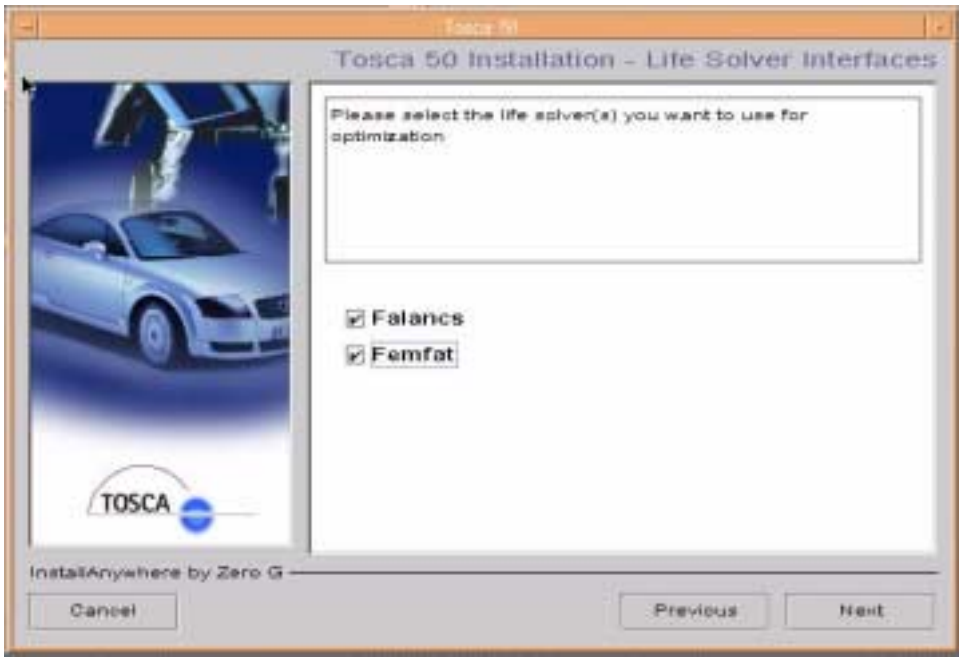
5.2.3 Selection of the solver interfaces

In the following dialog, the solver interfaces used with TOSCA are selected. The following solvers are supported in TOSCA 5.0:

- ABAQUS
- ANSYS
- I-DEAS
- MARC
- NASTRAN
- PERMAS
- FALANCS
- FEMFAT



The settings in the TOSCA configuration files are entered for the selected solver interfaces and the examples are copied into the installation directory. (see <tosca_install_dir>/<solver>/examples).



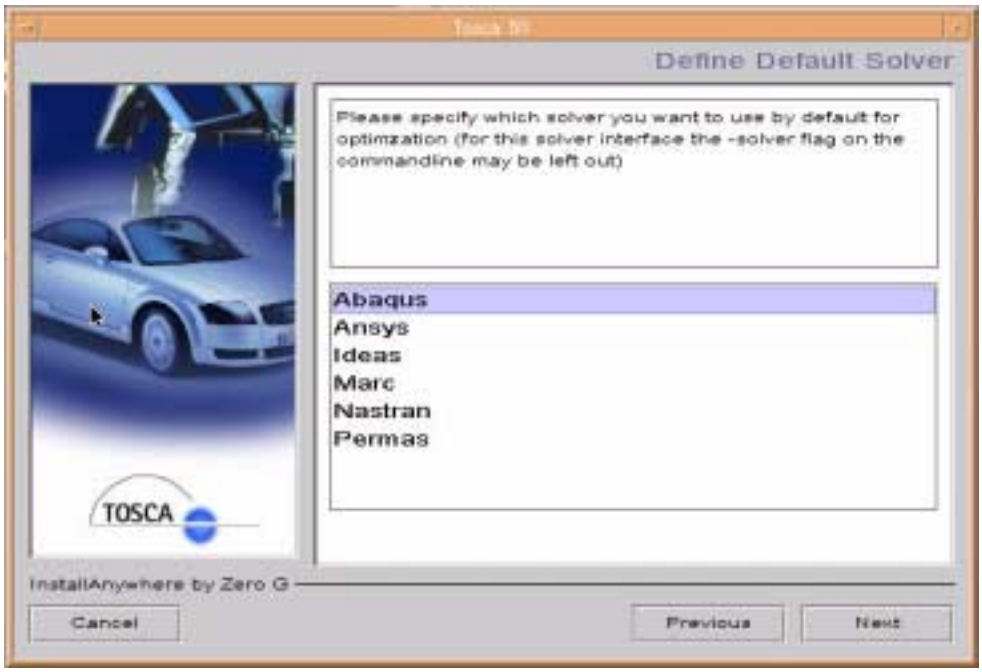
In addition to the finite element solvers, TOSCA supports the software FALANCS and FEMFAT for an integrated durability analysis in the optimization loop.

5.2.4 Selection of the default solver

The dialog box below shows how to specify the default solver for the TOSCA optimization run. The following line is added to the TOSCA configuration file <tosca_install_dir>/bin/tosca_ctrl.cfg :

```
if (!defined $fe_solver){$fe_solver = "<default_solver>"};
```

If the user starts TOSCA without any solver flag on the commandline (-solver ...), this default solver will be used for optimization.



5.2.5 Configuration of the solver calls

The following dialogs show how to specify the location of the solver executables. You can only search for products installed on the computer once your setup is running.

These settings are stored in the configuration files of TOSCA (`$tosca/bin/tosca_ctrl.cfg`) and may be modified manually at a later stage

For optimization with TOSCA, it is assumed that the finite element solvers are already installed and configured correctly.

TOSCA checks if the solver runs have been completed correctly during optimization. Therefore, no graphical start interfaces can be selected in this configuration (e.g. xnastran). The finite element solver run must not be added to a queuing system. If

you want to start the finite element solver run using a queuing system, some additional settings are required that are not covered by the TOSCA installation.

+ In the case of a common installation for Windows and Unix, the path names of the selected products must be entered for Windows and Unix. If you run the installation on a Windows computer you can select the installation path for the requested product with the file dialog box, but the installation path for Unix must be entered manually.

+ Depending on the selected finite element solver interface, several similar dialogs will appear. The configuration of the Abaqus interface is used as an example below.

5.2.5.1 ABAQUS

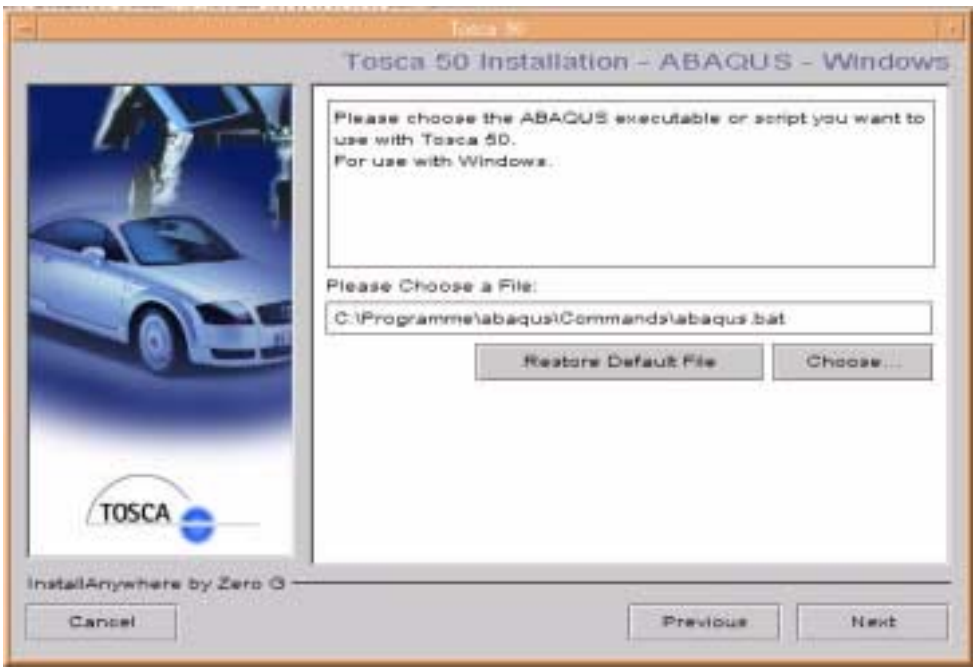
The differences between Windows and Unix dialogs are shown in the Abaqus interface configuration example below. If you have selected multiple solver interfaces, the order of appearance for the configuration dialogs is: first Windows, then Unix.

First, the Abaqus startscript has to be specified with the full path name

This is usually:

Windows: C:\ABAQUS\Commands\abaqus.bat

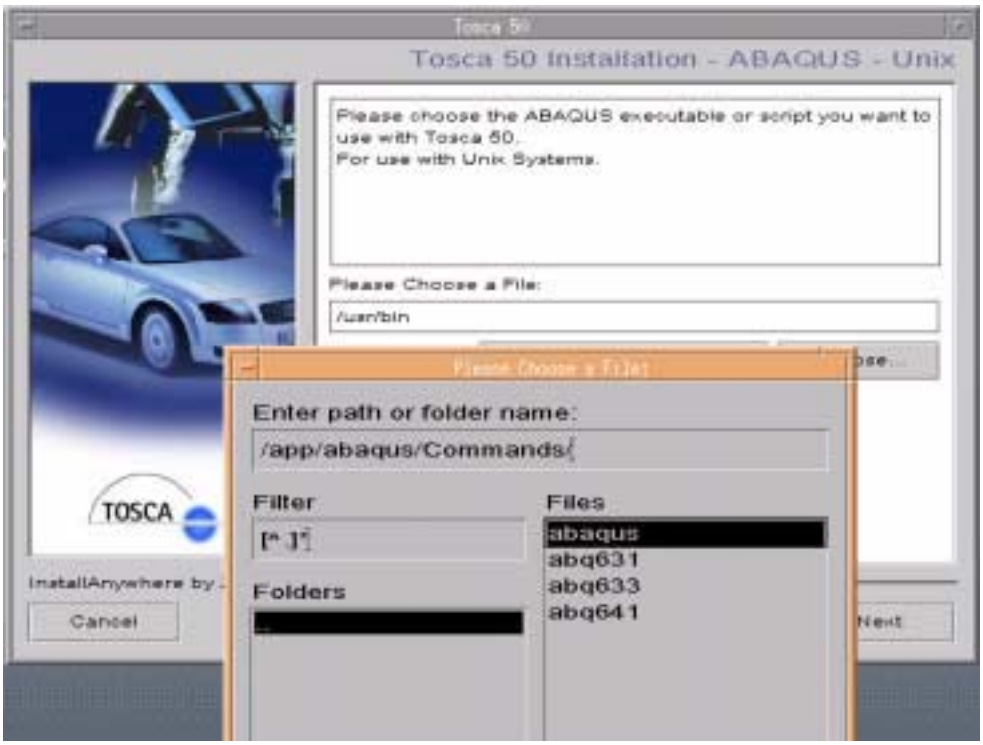
Unix: /usr/local/ABAQUS/Commands/abaqus



This example shows a Windows configuration dialog.

Version 5.0 Rev. 01 - 03.2004

The path to the Abaqus startscript must be entered - the selection dialog may not be used.



The Unix dialog only differs in the dialog title. If the installation is executed on a Unix platform, the file may also be selected using the 'Choose' Button.

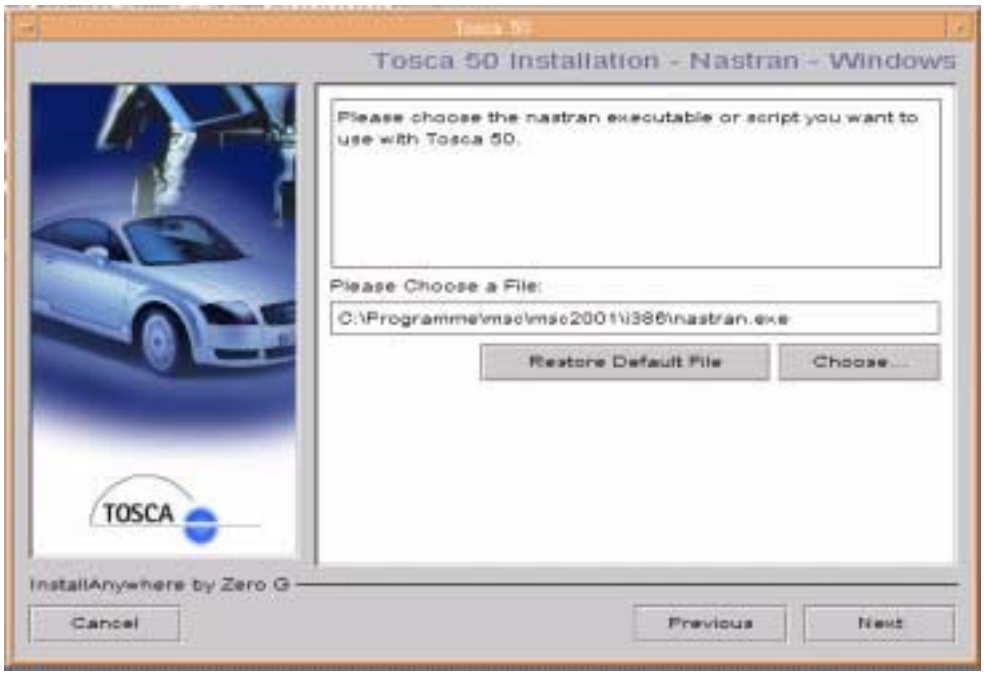
5.2.5.2 MSC.Nastan

Selection of the MSC.NASTRAN executable file or startscript.

The dialogs are similar to the Abaqus configuration dialogs.

\+ Notes for Windows users

On Windows systems, the `..../msc200x/i386/nastran.exe` executable file has to be used.



5.2.5.3 ANSYS

Selection of the ANSYS startscript.

In addition, the ANSYS product name has to be selected for the correct licensing of Ansys during optimization. If your product name is not listed, the product name ’others’ has to be selected and the product name may be entered manually.

5.2.5.4 I-DEAS

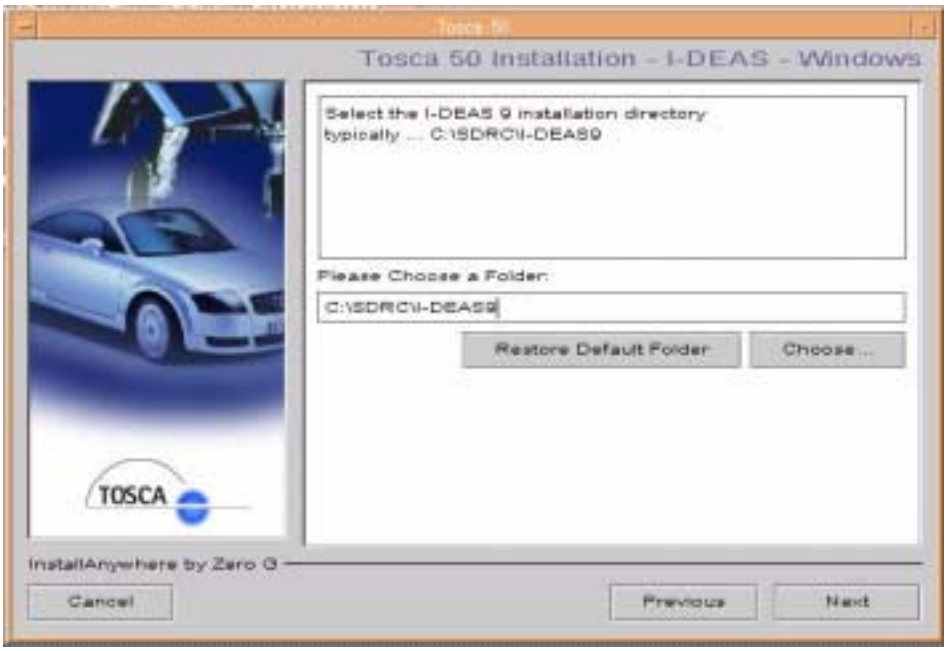
I-DEAS includes the Ideas Stand Alone Solver (SAS) which is used for optimization with TOSCA. First the installation directory of I-DEAS must be specified and then the startscript for the SAS solver has to be selected.

Special features with UNIX

The SAS solver must be completely installed and configured. Information regarding the SAS solver configuration settings can be found in the I-DEAS Installation guide for UNIX.

The startscript of the SAS solver is located in the following directory:

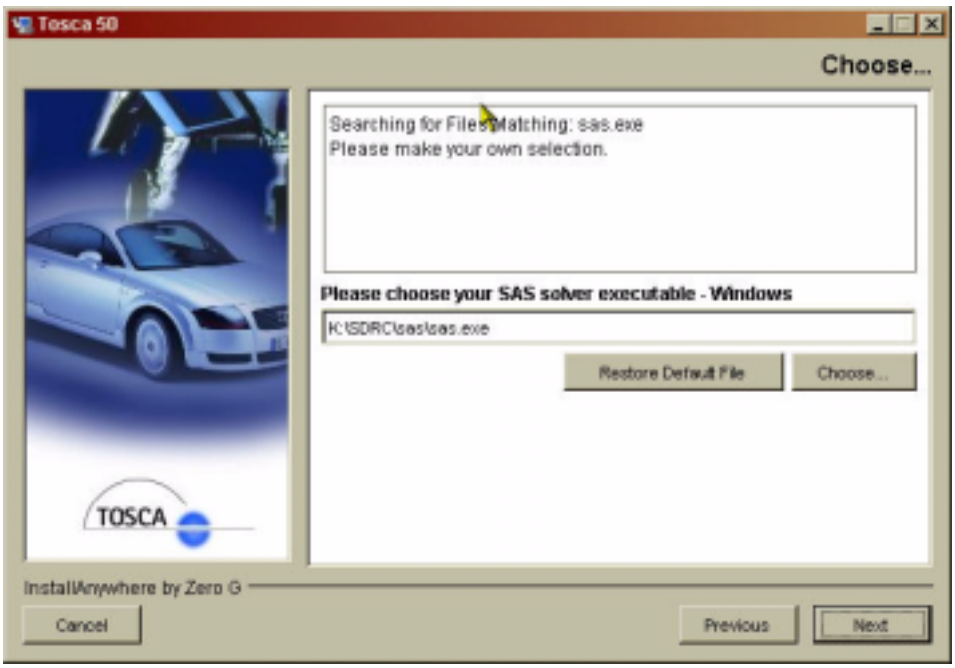
`.../SDRC/ideas9/sas/startsas.com`



Special features with Windows 2000/XP

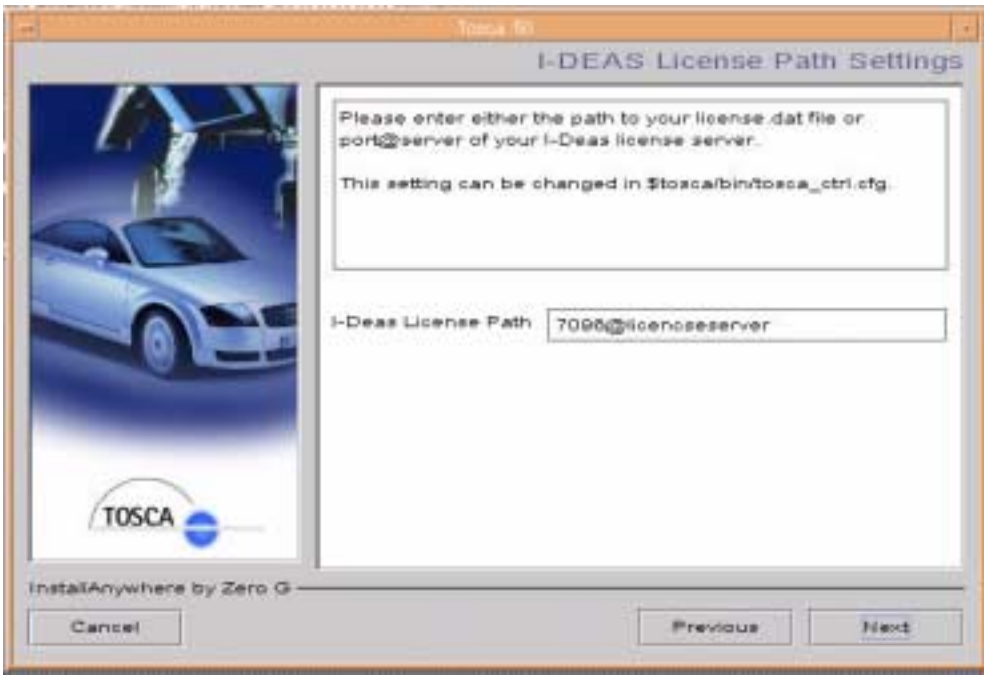
The solver executable file has to be selected on Windows systems.

`C:\SDRC\I-DEAS9\sas\sas.exe`



In addition, the license file of the I-DEAS installation must be copied into the $SDRC_INST/sas/SEC directory and renamed license.dat.

The SAS solver has to be configured as described in the I-DEAS installation guide for Microsoft Windows. TOSCA sets the necessary variables $SDRC_INSTL and the license variable $SDRC_MS9_LICENSE_FILE which correspond to the settings during the installation.

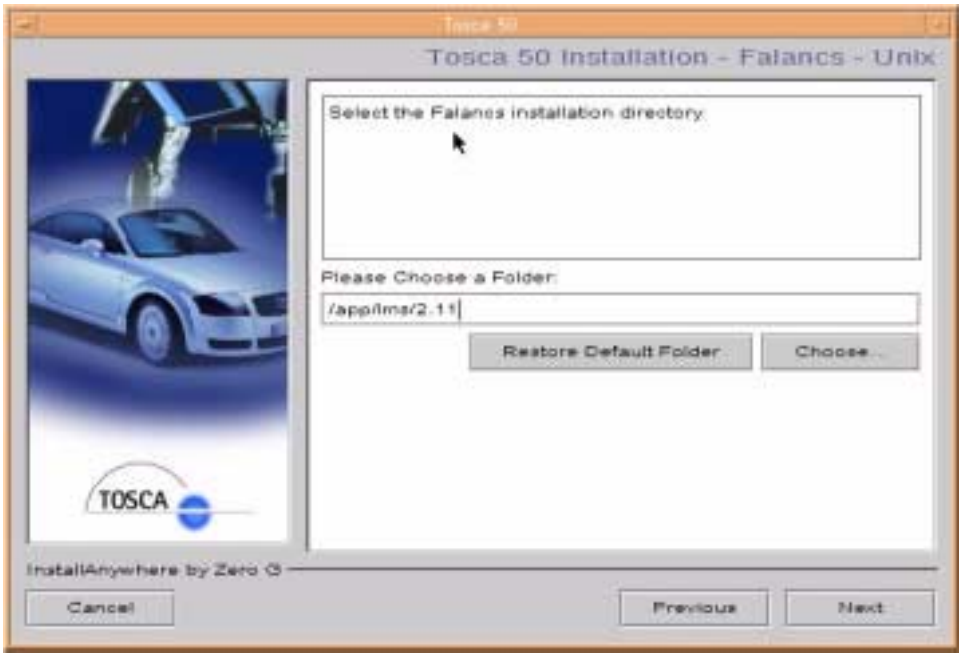


5.2.5.5 FALANCS

Falancs must run in batch mode for combined use with TOSCA. During installation, the FALANCS installation directory must be specified. This directory corresponds to the $TECWARE environment variable. If FALANCS is used in combination with TOSCA, the $TECWARE variable will be set in the optimization loop before FALANCS is called up.

During installation, an automatic search of the falbatch executable for the FALANCS batch run is made and can then be selected by the user.

FALANCS settings will be entered in the configuration file of the TOSCA execution control (`$tosca/bin/tosca_ctrl.cfg`).



5.2.5.6 FEMFAT

FEMFAT must run in batch mode if it is to be used in combination with TOSCA. During installation, the FEMFAT installation directory and the FEMFAT batch executable file must be selected.

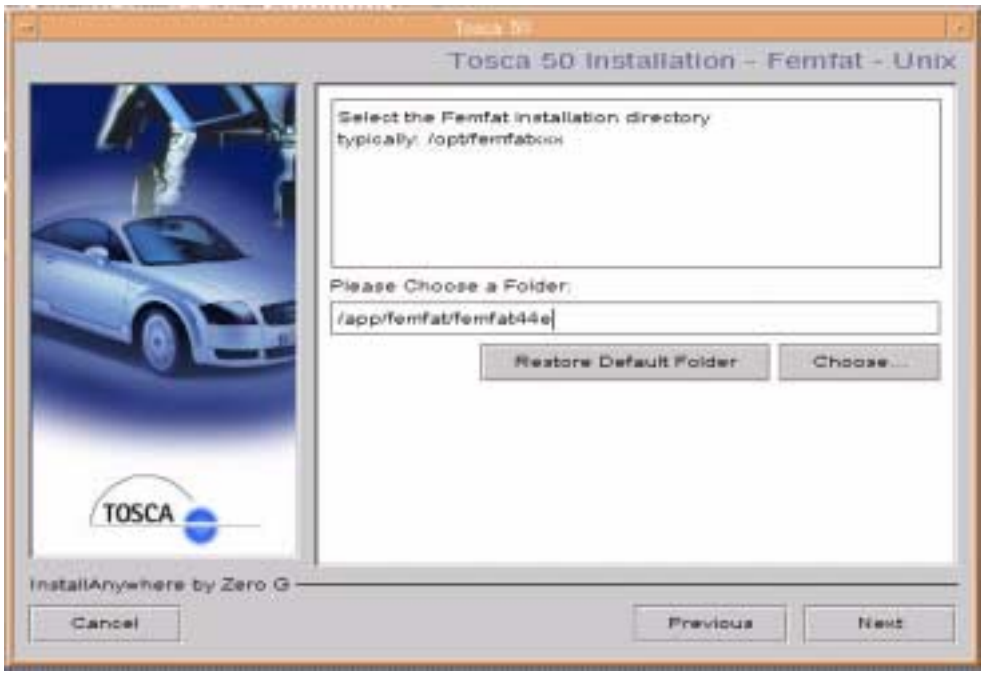
On Windows systems, the femfat.tbc file which is necessary for the FEMFAT batch call, must also be selected.

The specified settings will be entered in the configuration file of the TOSCA execution control (`$tosca/bin/tosca_ctrl.cfg`).

To load FEMFAT, the following entries are set:

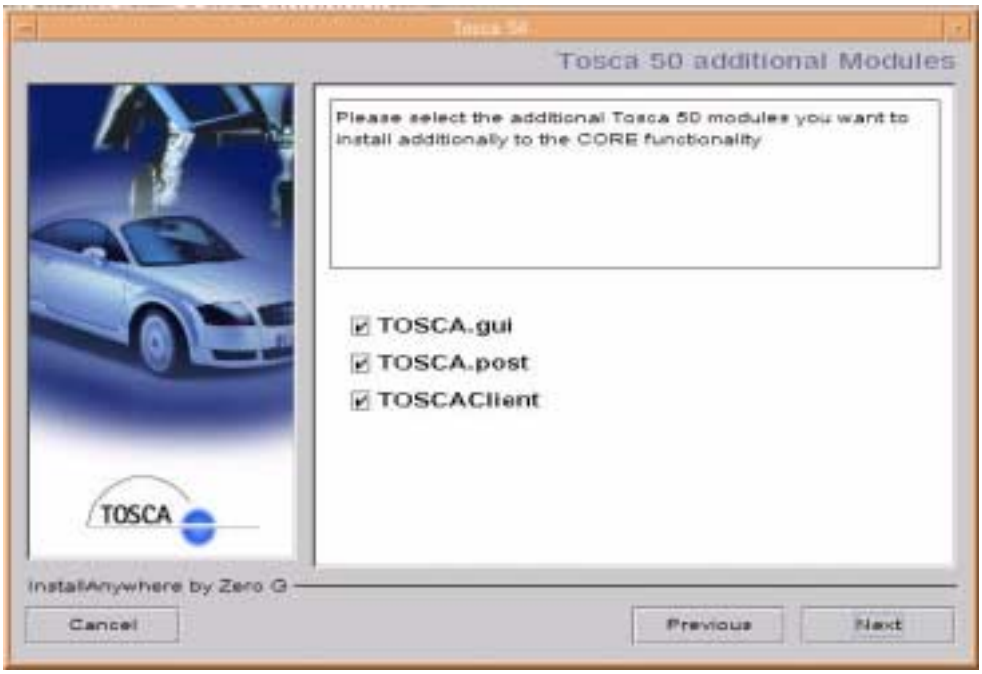
Windows: <femfat_install>/lib/dll/femfat.exe <femfat_install>/bin/bin/femfat.tbc

Unix: <femfat_install>/bin/femfat


Tosca 50 Installation – Femfat – Unix
Select the Femfat installation directory
typically: /opt/femfatxxx
Please Choose a Folder:
/app/femfat/femfat44e
Restore Default Folder
Choose...
TOSCA
InstallAnywhere by Zero G
Cancel
Previous
Next

5.2.6 Selection of the additional TOSCA modules

The additional TOSCA modules that should be installed may be selected in the following dialog:



5.2.6.1 TOSCA.gui

This module starts TOSCA, TOSCA.smooth, TOSCA.post, TOSCA.report and TOSCA.pre in a graphical environment.

The standard web Browser that should be used for the visualization of TOSCA report files must be selected.

5.2.6.2 TOSCA.post

TOSCA.post is used to generate postprocessing databases for the different postprocessing systems. Some of the postprocessor interfaces require the installation of additional components or need the entry of additional information.

ABAQUS/CAE und ABAQUS/Viewer

The ABAQUS/CAE interface makes use of some ABAQUS 6.4-1 runtime libraries. These will be copied to the $tosca/bin/<platform>/libHKS directory and is invoked

Version 5.0 Rev. 01 - 03.2004

by extending the library search path in the common_env.sh common environment script

MEDINA

The MEDINA interface requires a data element catalog dscat. Tosca will provide this and install it into the following location : $tosca/bin/<platform>

MSC.Patran

The path to the MSC.Patran executable file must be specified.

5.2.6.3 TOSCA.client

Calling up the TOSCA components is considerably easier if the TOSCA excutables in the installation directories are linked either to /usr/bin (or any other directory in your PATH Enviroment variable) or to the Windows start menu.

If you deselect this option the links will not be set.

5.2.7 Selection of the Web Browser

The TOSCA.report module can be used to generate a HTML page containing the optimization results. The use of the 'VIEW REPORT' button in TOSCA.gui shows the information regarding the path to the web browser. The browser executable has to be entered in this dialog.

5.2.8 Selection of the Acrobat Reader

The TOSCA manual is a pdf-document that can be visualized using the Adobe Acrobat Reader. The selected location of the Acrobat reader is automatically entered in all configuration files. The documentation of TOSCA can be seen by the user by selecting the HELP Button in TOSCA.pre and TOSCA.gui and on the command line by typing tosca -docu.

5.2.9 Generation of TOSCA links

The following links are generated in the program menu resp. the /usr/bin folder:

Windows:

- TOSCA 50 Command Shell
  TOSCA shell with all necessary environment variables set.

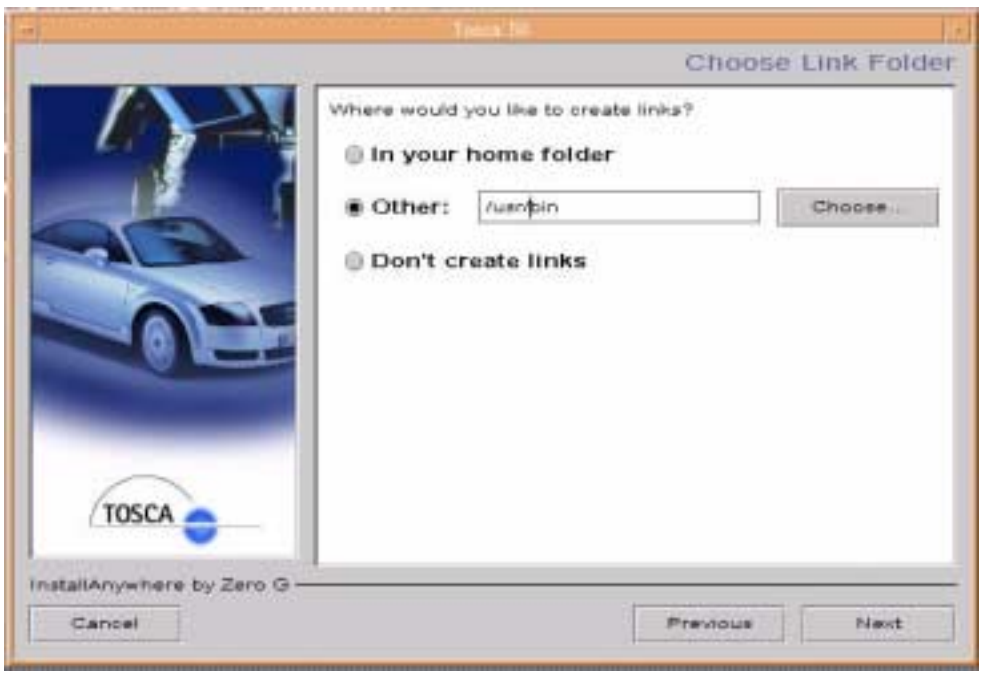
- TOSCA.gui
  Start of TOSCA.gui
- TOSCA Docu
  Link on TOSCA 50 Documentation

All TOSCA modules (e.g. tosca_smooth) may be started from the TOSCA 4.5.2 shell.

UNIX

- tosca50
  Start of TOSCA 5.0
- tosca50_gui
  Start of TOSCA.gui
- tosca50_post
  Start of the TOSCA.post module on the commandline
- tosca50_smooth
  Start of TOSCA.smooth on the commandline

The folder for the symbolic link generation is user-selectable.



+ Note

In the $tosca/bin directory, the scripts InstallToscaLinks.sh resp. InstallToscaLinks.cmd can be found which can generate links on the client platform in case of a network installation.

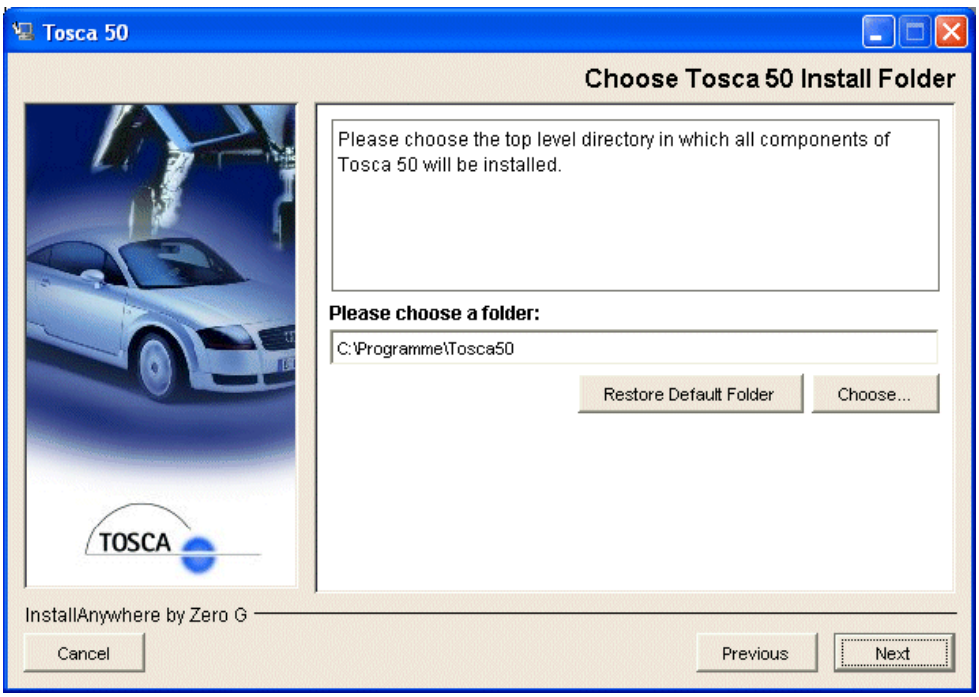
5.3 TOSCA Personal Installation

The TOSCA Personal Edition is a TOSCA Version that only runs on Windows systems. The TOSCA 50 PE supports the following finite element solvers:

- MSC.visual Nastran for Windows
- Ansys
- I-DEAS

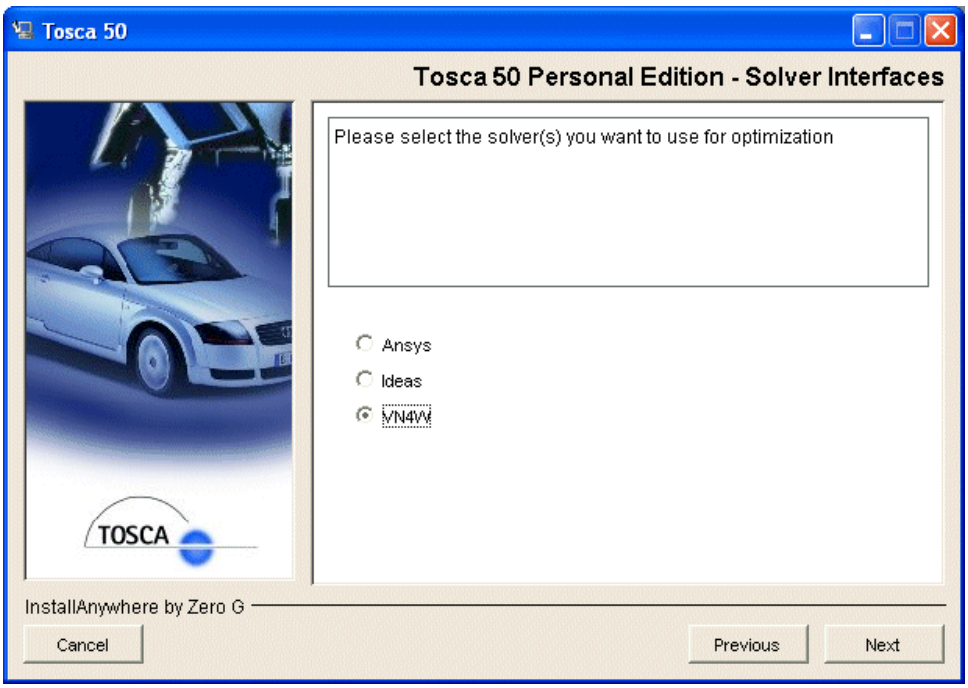
5.3.1 Selection of the installation directory

All TOSCA files will be extracted to the selected directory.



5.3.2 Selection of the solver interfaces

The solver interfaces for the use of TOSCA are selected with the following dialog.



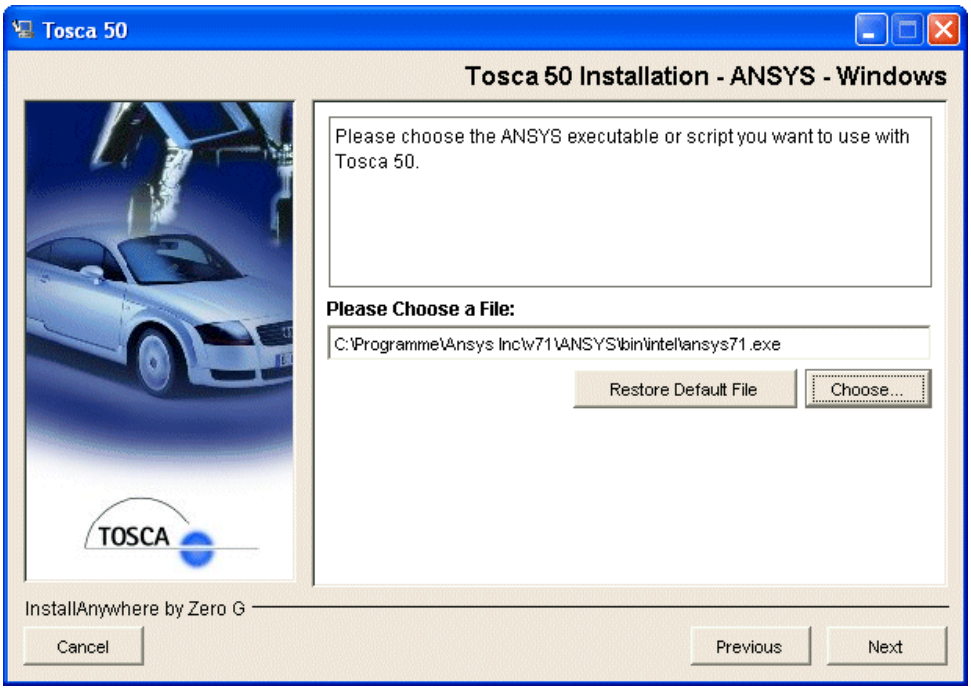
Version 5.0 Rev. 01 - 03.2004

5.3.3 Configuration of the solver calls

5.3.3.1 Ansys

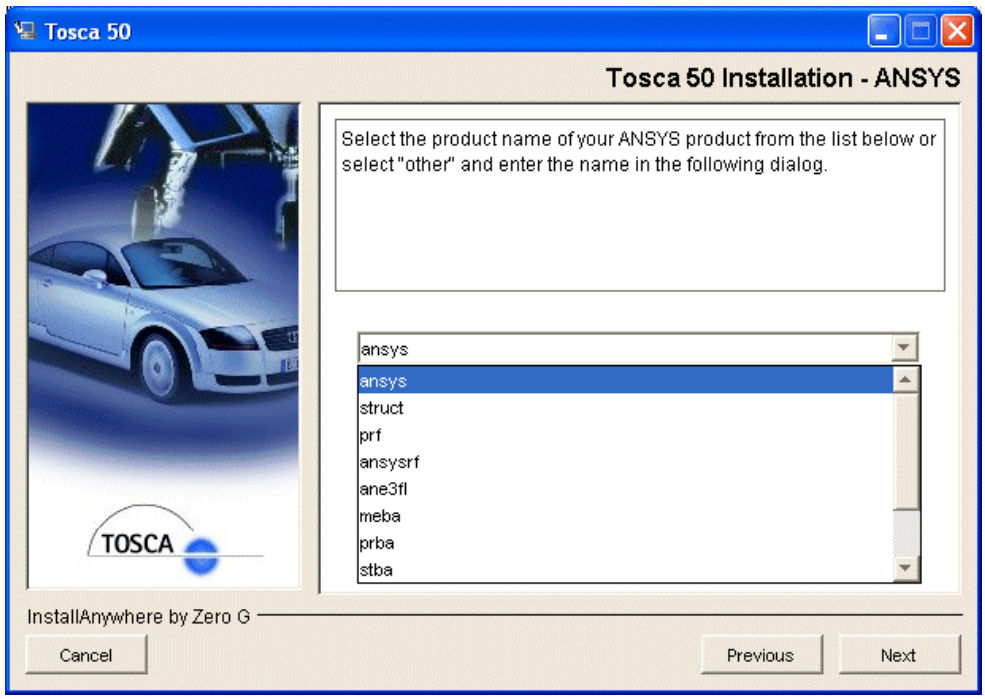
Selection of the solver executable

If the Personal Edition for Ansys is installed, the path to the Ansys batch executable has to be selected.



Selection of the ANSYS product

The ANSYS product name also has to be selected for the correct licensing of Ansys during optimization. If your product name is not listed, the product name ’others’ has to be selected and the product name may be entered manually.



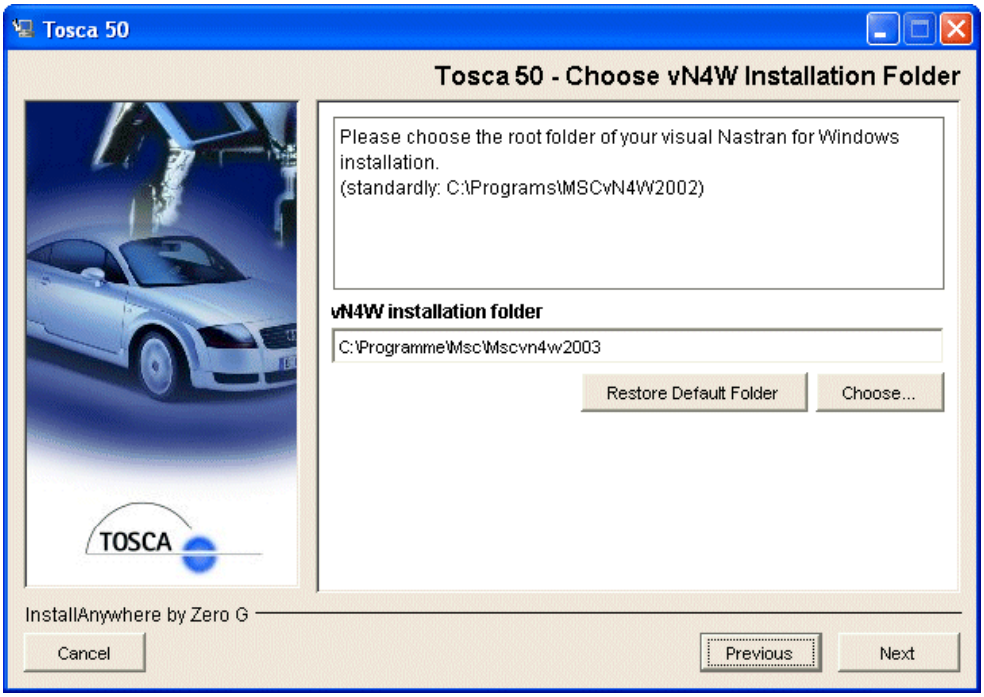
5.3.3.2 MSC.visual Nastran for Windows (vN4W)

Selection of the vN4W installation directory

The installation directory of the MSC.visual Nastran for Windows installation has to be selected. Per standard the vN4W application is installed in

```
C:\Program files\MSCvN4W2003
```

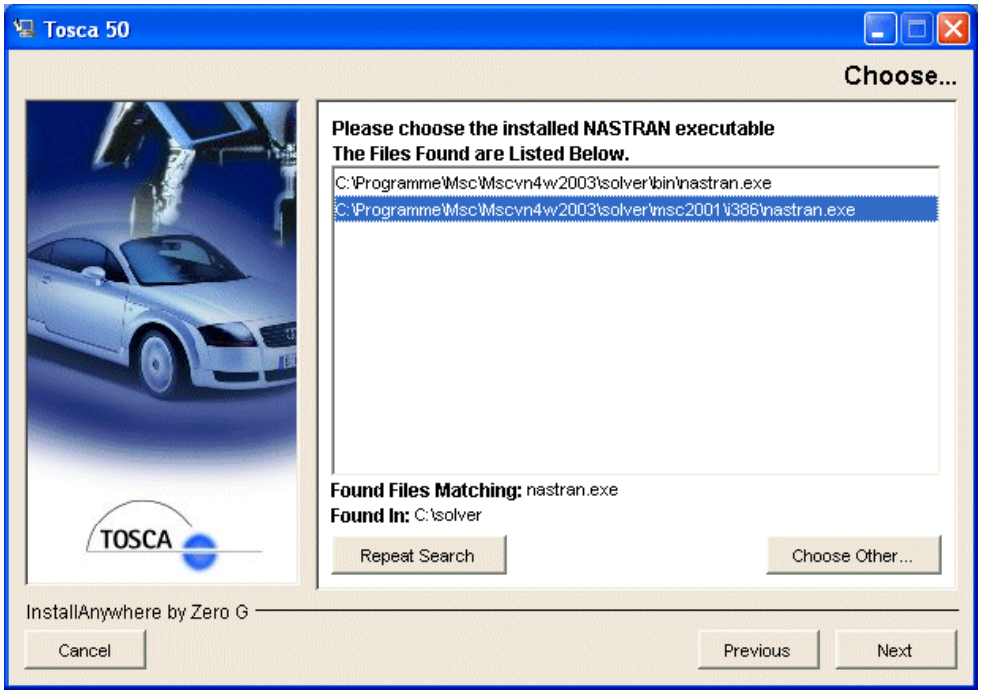
During installation, a search is made for the necessary configuration files of vN4W in the specified installation directory.



Selection of the solver executable

The next step is to select the Nastran executable that should be used during optimization. The existing Nastran executables in the above selected vN4W

installation are presented in a list and the Nastran executable in the i386 directory should be selected.

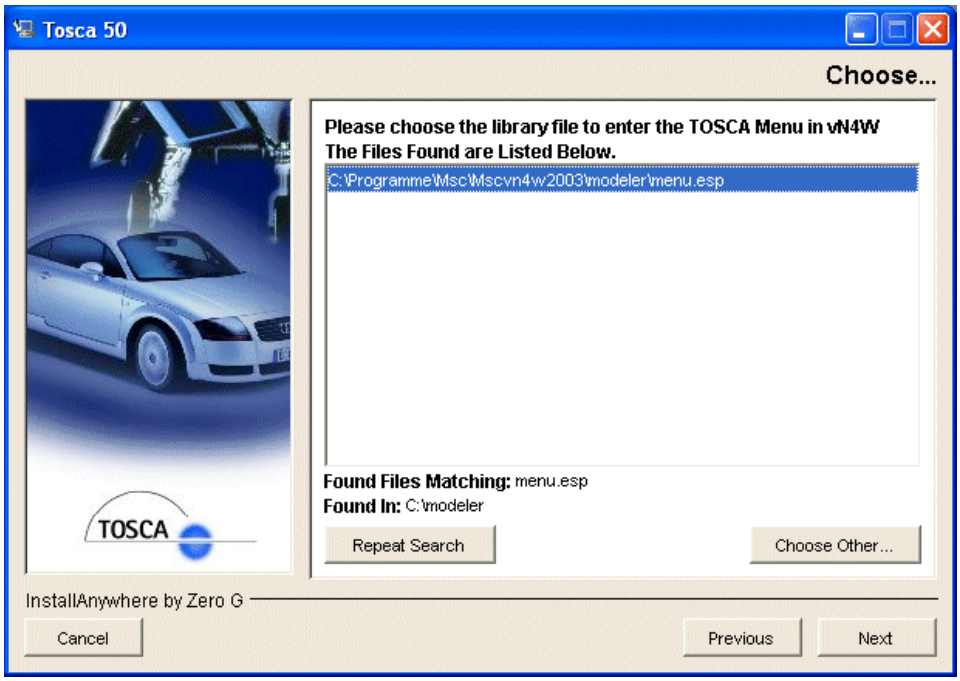


Register the TOSCA menu in the vN4W menubar

All necessary scripts for pre and postprocessing of an optimization task in combination with vN4W are installed in the `$tosca/vN4W/gui` directory. To be able to access these scripts directly from vN4W, the configuration of the vN4W installation has to be enhanced.

Per standard, the menu.esp file in the modeler directory of the vN4W installation contains the menu settings for vN4W. The TOSCA menu entries are added to this file. The file has to be selected during the installation of TOSCA. If the menu.esp file does not exist in the modeler directory of the vN4W installation, TOSCA will generate this file.

Using the button “CHOOSE OTHER“, the TOSCA menu may be added to an arbitrary file. This file may be selected as a menu file in the ’`Library Preferences` -> `Menu`’ Dialog in vN4W. If a TOSCA menu entry already exists in the menu.esp file (e.g. from a previous TOSCA installation), the menu.esp should be edited manually to delete the non-used entries.



The following entries are added to the menu.esp file:

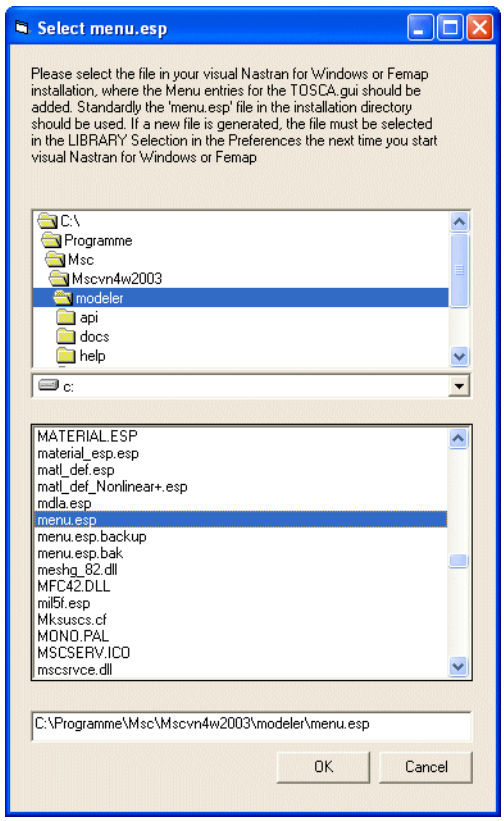
```
POPUP "TOS&CA"
  BEGIN
  MENUITEM"TOSCA&Wizard","C:\program files\Tosca50pe\vn4w\gui\tosca_wizard.exe"
  MENUITEM SEPARATOR
  MENUITEM"&Manage Groups","C:\pprogram file\Tosca50pe\vn4w\gui\manage_groups.exe"
  MENUITEM"&Export TOSCA Groups","C:\program file\Tosca50pe\vn4w\gui\export_groups.exe"
  MENUITEM SEPARATOR
    MENUITEM "Start TOSCA.&gui","C:\program file\Tosca50pe\bin\tosca_gui.exe"
  MENUITEM SEPARATOR
  MENUITEM "Import TOSCA &Results","C:\program file\Tosca50pe\vn4w\gui\read_results.exe"
  MENUITEM "&Topo Postprocessing","C:\program file\Tosca50pe\vn4w\gui\topo_post.exe"
END
```

Generation of the vN4W menu entries after installation

Registration of the TOSCA menu in the vN4W menu bar can also occur after installation has been completed. This actualisation may be necessary for two reasons

1. Actualisation of the vN4W version on the computer after the installation of TOSCA has been completed and the menu.esp file is no longer present
2. Multiple users are using the same TOSCA version in a network installation from different computers. Actualisation of the menu entries has to be performed on each computer.

After the ’generate_menu’ module is started (’C:\Program files\Tosca50_pe\vn4w\generate_menu’ resp. in the start menu) the menu.esp file in the vN4W has to be selected where the menu settings of the TOSCA modules are added.



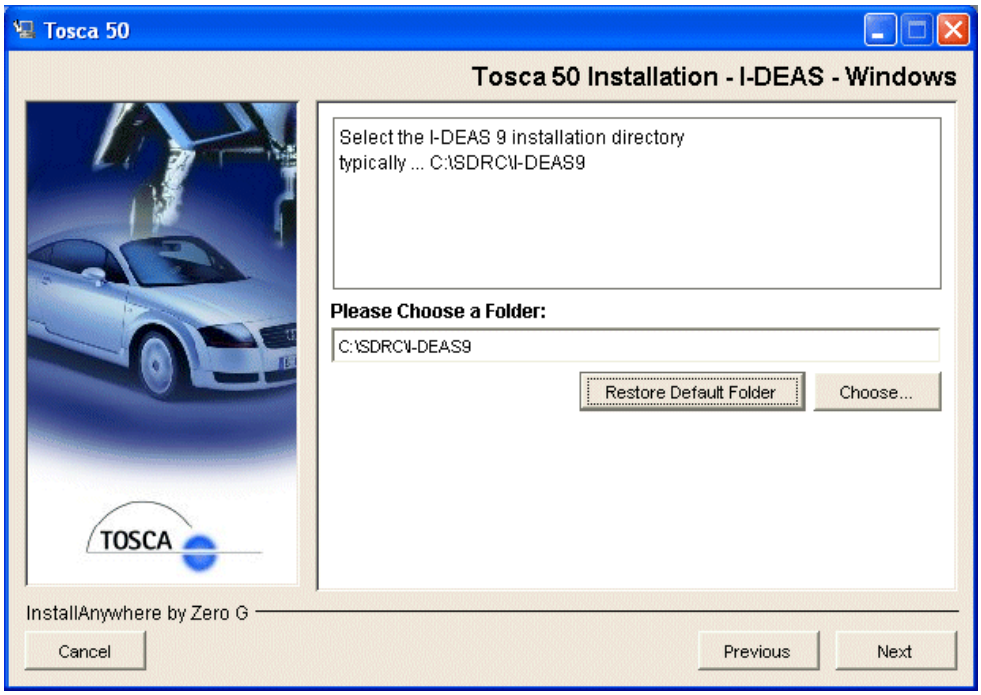
5.3.3.3 I-DEAS

Selection of the I-DEAS installation directory

The installation directory of the I-DEAS installation must be selected. Per standard the I-DEAS application is installed in

```
C:\Program files\MSCvN4W2003
```

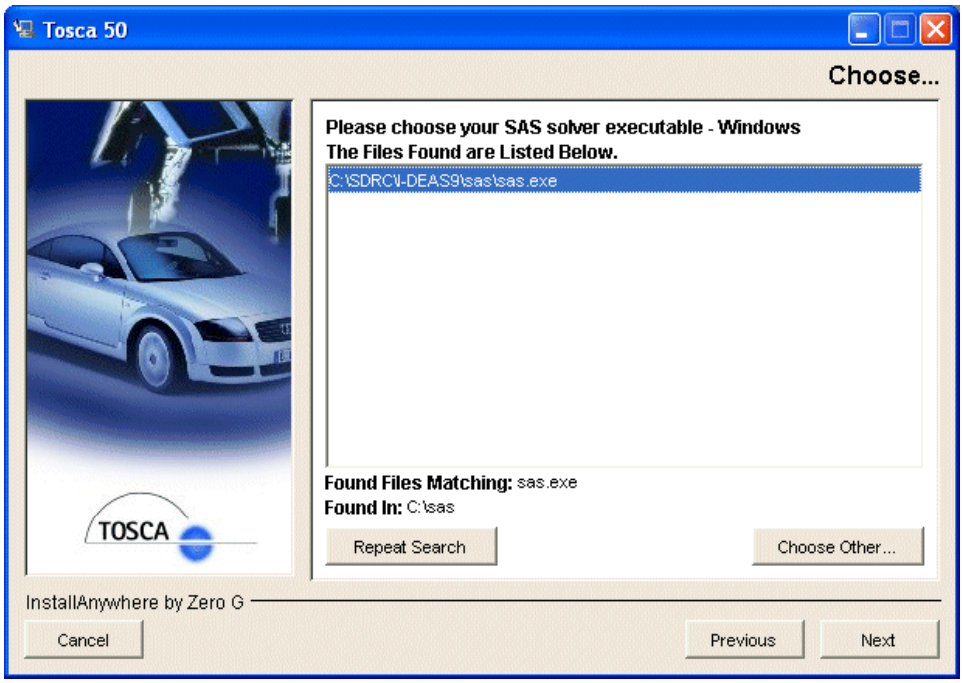
During installation, a search for the necessary configuration files for the I-DEAS interface is made in the specified installation directory.



Selection of the SAS solver executable

The next step is the selection of the SAS solver executable that should be used during optimization. The existing SAS solver executables in the above selected I-DEAS

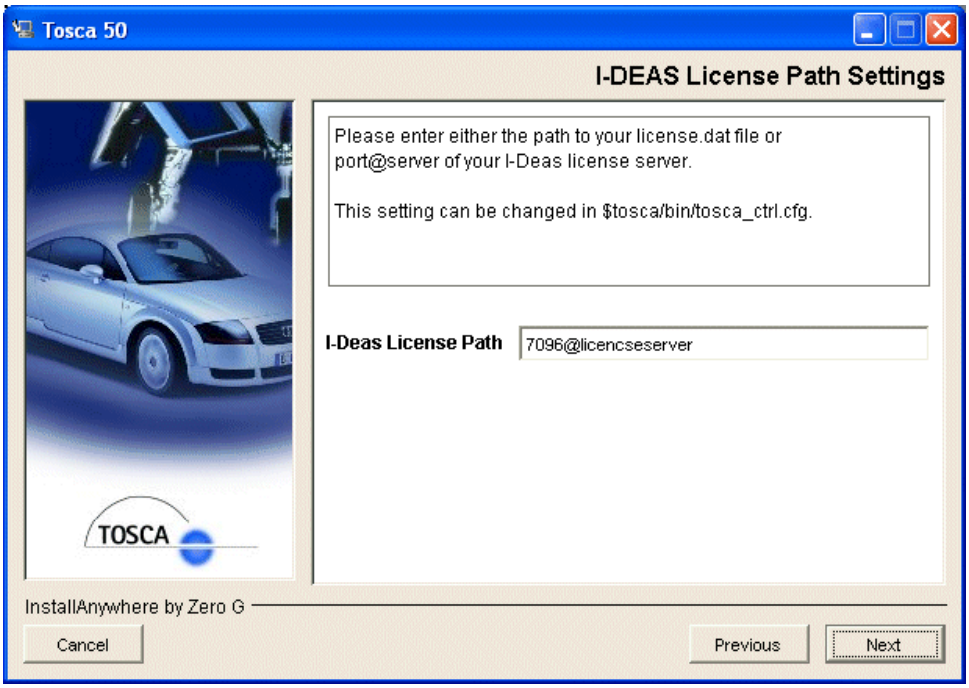
installation are presented in a list. The selected SAS executable is added to the TOSCA configuration file tosca_ctrl.cfg.



Selection of the I-DEAS license file

The I-DEAS license file has to be copied into the $SDRC_INSTL/sas/SEC directory and renamed license.dat.

TOSCA sets the environment variables $SDRC_INSTL and the license file variable $SDRC_MS9_LICENSE_FILE variable during optimization.



5.3.4 Selection of the WEB BROWSER

The TOSCA.report module can be used to generate a HTML page containing the optimization results. The 'VIEW REPORT' button in TOSCA.gui provides information regarding the path to the web browser. The browser executable has to be entered in this dialog.

5.3.5 Selection of the Acrobat Reader

The TOSCA manual is a pdf-document that can be visualized using the Adobe Acrobat Reader. The selected location of the Acrobat reader is automatically entered in all configuration files. The documentation of TOSCA can be seen by the user by selecting the HELP Button in TOSCA.pre and TOSCA.gui and on the command line by typing tosca -docu

5.3.6 Generation of the TOSCA links

The following links are generated in the program's menu

- generate_menu (only vN4W Interface)
  Generating the TOSCA menu in the vN4W environment

- TOSCA 50 Command Shell
  TOSCA shell with all necessary environment variables set.
- TOSCA.gui
  Start of TOSCA.gui
- TOSCA Docu
  Links on the TOSCA 50 documentation

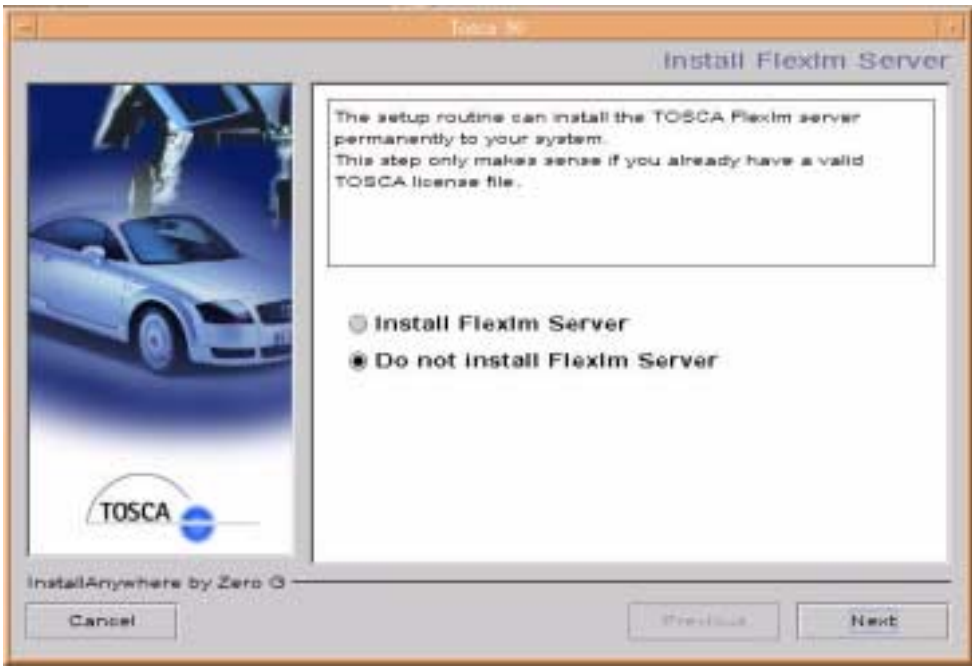
All TOSCA modules (e.g. tosca_smooth) may be started from the TOSCA 4.5.2 shell.

5.4 Installation of the FlexLM license server

Installation of the FlexLM license server may be performed during the installation of TOSCA. The FlexLM server installation may also be performed after the installation of TOSCA has been completed. All files required will be copied in the installation directory of TOSCA and the set up of the license server may be carried out manually at a later time.

+ The automatic server installation writes files in the system configuration of the installation computer and configures FlexLM to be executed as actual user (typically 'root' at installation time).
If you prefer to install the Flex Server manually you can modify the startFEDFLEXLM script and adapt it to your needs. Additional information may be found in the FlexLM enduser manual and the documentation of your operating system.

On Windows systems, an additional link in the TOSCA program's menu is generated that allows the start of the FlexLM Windows administration user interface 'lmtools.exe'. This graphical user interface allows the comfortable set up and control of the FlexLM license manager.


Install Flexlm Server
The setup routine can install the TOSCA Flexlm server permanently to your system.
This step only makes sense if you already have a valid TOSCA license file.
Install Flexlm Server
Do not install Flexlm Server
TOSCA
InstallAnywhere by Zero G
Cancel
Next

At this point in time, TOSCA is completely installed and configured.

The components required for the FlexLM installation must contain the directory structure that is described in the following chapter. In addition, the structure of the startscripts needs to fulfil the requirements as described below.

If the requirements are not fulfilled, FE-Design recommends that the FlexLM server be installed manually.

5.4.1 Automatic installation of the FlexLM server

First, the following files are copied to the TOSCA installation directory:

Tosca50	TOSCA installation directory			
	startFEDLEXLM			Unix startscript FlexLM server
	license	license.dat		license file template
	flexlm	Docu	enduser.pdf	Flexlm User Manual
		os_<platform>	lmgrd	License Server
			FED	FED Vendor Daemon
			lmutil	FlexLM utilities
			tosca_license	Installation and test of the TOSCA license server
			tosca_client	Test module for the check out of the TOSCA licenses.

The tosca_license module may execute the installation of the FlexLM server in the corresponding platform.

5.4.2 Required structure of the boot-scripts on the UNIX target system

On UNIX systems, the startFEDFLEXLM script is copied into the boot structure of the corresponding UNIX system and added to the runlevel configuration.

- AIX
  startFEDFLEXLM is copied to /etc/rc.d/rc2.d/S95startFEDFLEXLM
- HP-UX
  startFEDFLEXLM is copied to /sbin/init.d
  /sbin/init.d/startFEDFLEXLM is linked to /sbin/rc2.d/S95startFEDFLEXLM
- IRIX
  startFEDFLEXLM is copied to /etc/init.d
  /etc/init.d/startFEDFLEXLM is linked to /etc/rc2.d/S95startFEDFLEXLM
- Linux
  startFEDFLEXLM is copied to /etc/init.d

/etc/init.d/startFEDFLEXLM is linked to
/etc/init.d/rc2.d/S95startFEDFLEXLM

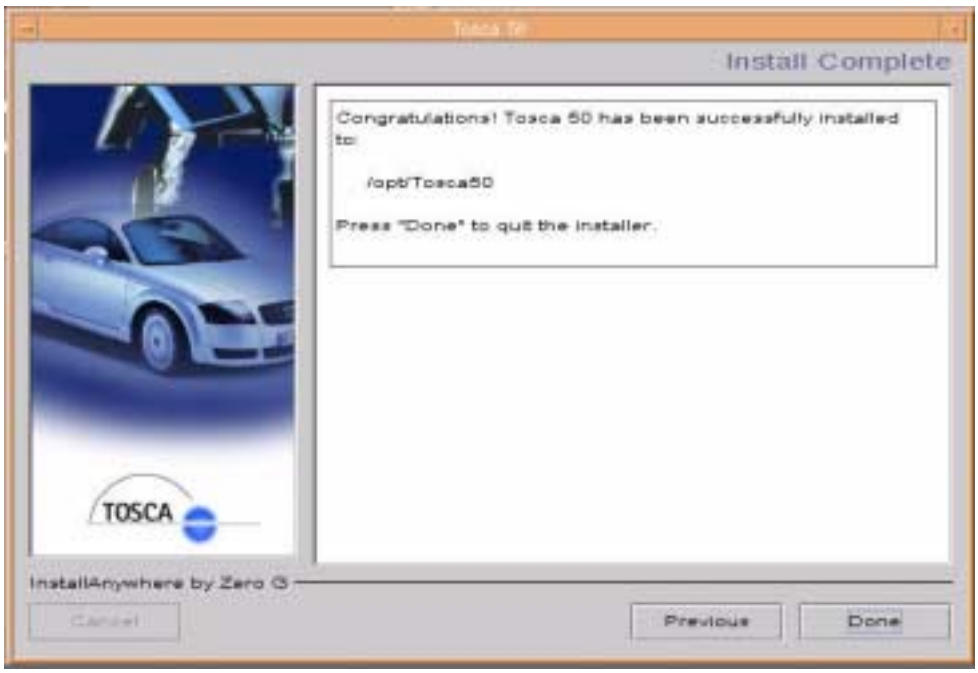
- Solaris
  startFEDFLEXLM is copied to /etc/init.d
  /etc/init.d/startFEDFLEXLM is linked to
  /etc/rc2.d/S95startFEDFLEXLM

5.4.3 Installation on Windows systems

On Microsoft Windows systems the FlexLM program installs.exe is used to configure the license server as a system service. The configuration and administration can be easily made with lmtools.

5.5 Summary

The selected components are summarized at the completion of installation procedures. The TOSCA 5.0 files are copied to the hard disk and the necessary configuration settings are carried out.



The final dialog confirms the succesful installation of TOSCA Version 5.0

6 Licensing

6.1 FlexLM in general

FlexLM offers two different types of licenses: node lock licenses that are only valid on enabled computers, and network licenses that are accessible from all computers in a network.

In this chapter, only a description of the special features pertaining to the FE-Design implementation and the delivered tools will be given. A detailed description of the Flex License Manager may be found in the enduser documentation of Macrovision (see CD/info/enduser.pdf).

A license server consists of 3 components:

- lmgrd ... license server
- FED ... FE Design Vendor Daemon
- license.dat ... license file

The FlexLM server 'lmgrd' is the general part of the license server and carries out the licensing. The lmgrd process starts the vendor daemon (FED). This process depends on the software product and controls the communication between FlexLM and the software application.

The license.dat file contains information on your configuration and the license keys for TOSCA.

6.2 Preparation

First you have to decide which computer is to be used as the FlexLM server. A CPU should be selected on which the configuration will not be changed.

To install FlexLM you may use the TOSCA installation program or copy the delivered tar archive for the corresponding platform.

6.3 Determination of the FlexLM hostid

The FlexLM host ID is required in order to apply for a license. It contains a unique id of your license server. There are two possibilities to determine the host ID:

If you perform the installation of the FlexLM server using the TOSCA installation procedure the file

TOSCA_LICENSE_REQUEST_FORM.txt

is generated in the $tosca/license directory. This file contains all information regarding the license server and the FlexLM host ID.

If you execute the installation manually, you may use the lmhostid or the lmutil program to determine the host ID.

The TOSCA_LICENSE_REQUEST_FORM.txt or the output of lmhostid has to be sent to FE-Design to obtain the relevant license.

6.4 FlexLM Software

Macrovision provides the ’lmutil’ utility for the administration of the FlexLM license server. A detailed description may be found in the enduser Manual of FlexLM. You will find a troubleshooting chapter at the end of the TOSCA installation Guide.

6.5 TOSCA tools

FE-DESIGN provides two additional utilities

- tosca_license
  this program is located in <tosca50>/flexlm/os_<platform> and provides the following options :
  tosca_license [-c[LICENSE_PATH]][-i][-r][-t]
  -i ... installs the TOSCA Flexlm Server
  -r ... removes the TOSCA Flexlm Server
  -t ... tests the TOSCA Flexlm Server by calling lmdiag, lmstat und tosca_client
  -c ... defines the license path

  The TOSCA FlexLM server has to be installed in the predefined file structure during the TOSCA installation in order to use the tosca_license.e component.
  tosca_license is only used for the installation of the license server.
  During deinstallation of the TOSCA FlexLM server, hard coded path settings to the startscripts and the RC structure are used to avoid the deinstallation of other products. The only disadvantage is that the Flex server of another product may be

shut down during the lmdown call if the license.dat file contains references to this Flex server.

tosca_client
tosca_client is located in the installation directory of the FlexLM programs and in the $tosca/bin/os_<platform> directory.
tosca_client may also be called up by the user to check if the TOSCA licenses are correct. If no license is available, tosca_client is queued until a license can be examined. To get a list of all users which hold a license, the FlexLM utility <lmutil lmstat -a> should be used.

tosca_client LIZENZNAME [LIZENZVERSION] [-c [LICENSE_PATH]]
Bsp:
$ ./tosca_client TOSCA_CORE -c 6666@fed-ex2

License PATH is set to: 6666@fed-ex2
License job initialized
TOSCA_CORE 5.0 checked out on 6666@fed-ex2

6.6 Installation of a “node-lock“ license

If TOSCA is to be used with a node locked license, the license.dat file that is sent by FE-Design has to be copied into the $tosca/licens directory.

6.7 Installation of a license server

To be able to start the license server, the following entries in the license.dat file have to first be made.

The Header must contain the correct SERVER and if necessary the VENDOR line. These are the only lines that may be modified by the user.

A SERVER line has the following entries:

```
SERVER <Server Name> <flexlmid> <Server TCP/IP Port>
```

Note: TOSCA uses the standard port 27000. It is useful to define the port in such a way that TOSCA does not conflict with other FlexLM authorized software applications.

The template contains the following line

```
SERVER YourLicenseServerName 00b0d0090b85
```

To start the license server in the machine fed-pc14 and the TCP/IP Port 1234 you have to modify the following entries:

```
SERVER fed-pc14 00b0d0090b85 1234
```

A VENDOR line has the following entries:

```
VENDOR VENDOR_DAEMON_NAME PATH_TO_VENDOR_DAEMON (additional
OPTIONS if necessary)
```

The template contains

```
VENDOR FED
```

If the vendor daemon executable is not installed in the same directory as the lmgrd license server the complete path has to be specified:

```
VENDOR FED c:\flexlm\bin\FED.exe
```

All other lines (FEATURE) must not be edited !!

```
FEATURE TOSCA_CORE FED 5.0 31-dec-2003 1 VENDOR_STRING="FE-Design \
        GmbH 2003;" ISSUED=9-1-2003 START=26-jun-2003 SIGN="1F8A D8A0 \
```

Now, the license server may be started using

```
lmgrd -c /path_to_license.dat
```

e.g. on a Unix system:

```
lmgrd -c /opt/Tosca50/license/license.dat
```

or on a Windows system:

```
lmgrd -c c:\Programme\Tosca50\license\license.dat
```

Now the license server is running. It has to be restarted manually after a systems reboot.

On a Windows System the license server may be added as a system service with the following commandline:

```
installs -c <license file path> \
         -e <path to lmgrd.exe> \
         -l <log file path> \
         -n <service name> \
         [-k <lmgrd parameters>]
```

Or the Windows interface lmtools.exe may be used.

On a Unix system a startscript has to be generated that must be added to the boot sequence of the operating system. Further information can be found in Chapter 6 of the FlexLM enduser documentation.

6.8 Installation of redundant license servers

If redundant license servers are to be installed you have to determine all host ID's of the CPU's in use it. In this case, you will receive an enhanced license file from FE-Design. A detailed description on the use of redundant license servers may be found in the FlexLM enduser manual in Chapter 4.

6.9 Configuration of TOSCA

The license path has to be transferred to a FlexLM authorized application. Two options are available:

1. TOSCA searches the FED_LICENSE_FILE variable in the process environment. This environment variable may contain the path to a license.dat file or the definition of a license server.

   e.g.

   ```
   FED_LICENSE_FILE=/opt/Tosca50/license/license.dat
   ```

   or

   ```
   1234@fed-pc14 (see example above)
   ```

2. TOSCA searches the file license.dat in the $tosca/license directory which contains the necessary information

The installation sets the FED_LICENSE_FILE variable in the following files:

Unix: common_env.sh

Windows: common_env.cmd

This implies the setting of the FED_LICENSE_FILE variable in the process environment. The specified path is the first of the FlexLM license path.

6.10 Test of the installation

A check as to whether the license server functions correctly may be made with the FlexLM utilities. The lmtools.exe utility can be used on Windows systems. Some commandline tools (e.g. Unix systems) are also available.

the commandline

```
lmdiag -c <path_to_license.dat>
```

may be used to check if the installed licenses are available - i.e. if they can be checked over the network environment.

A detailed description of all other FlexLM utilities can be found in the enduser documentation.

6.11 Note

FlexLM saves information regarding the licensing order in the running server processes. This leads to a loss of information in case of a license server shutdown. After a restart of the vendor daemon, the license server has no information about the licenses which have been checked out

This may lead to problems in a situation of concurrence where a running TOSCA process and a queued or newly-started process are looking for the same license.

When starting, TOSCA determines the running time of the FED process. If the vendor daemon is running less than 150 sec, TOSCA waits 150 sec until the license has been checked. This ensures that all running processes restore the connection to the licensed server before the new license requests are served.

Version 5.0 Rev. 01 - 03.2004

7 Test of the installation

After the TOSCA license has been successfully installed, TOSCA Version 5.0 may be executed and tested using a test example.

The test examples may be accessed as follows:

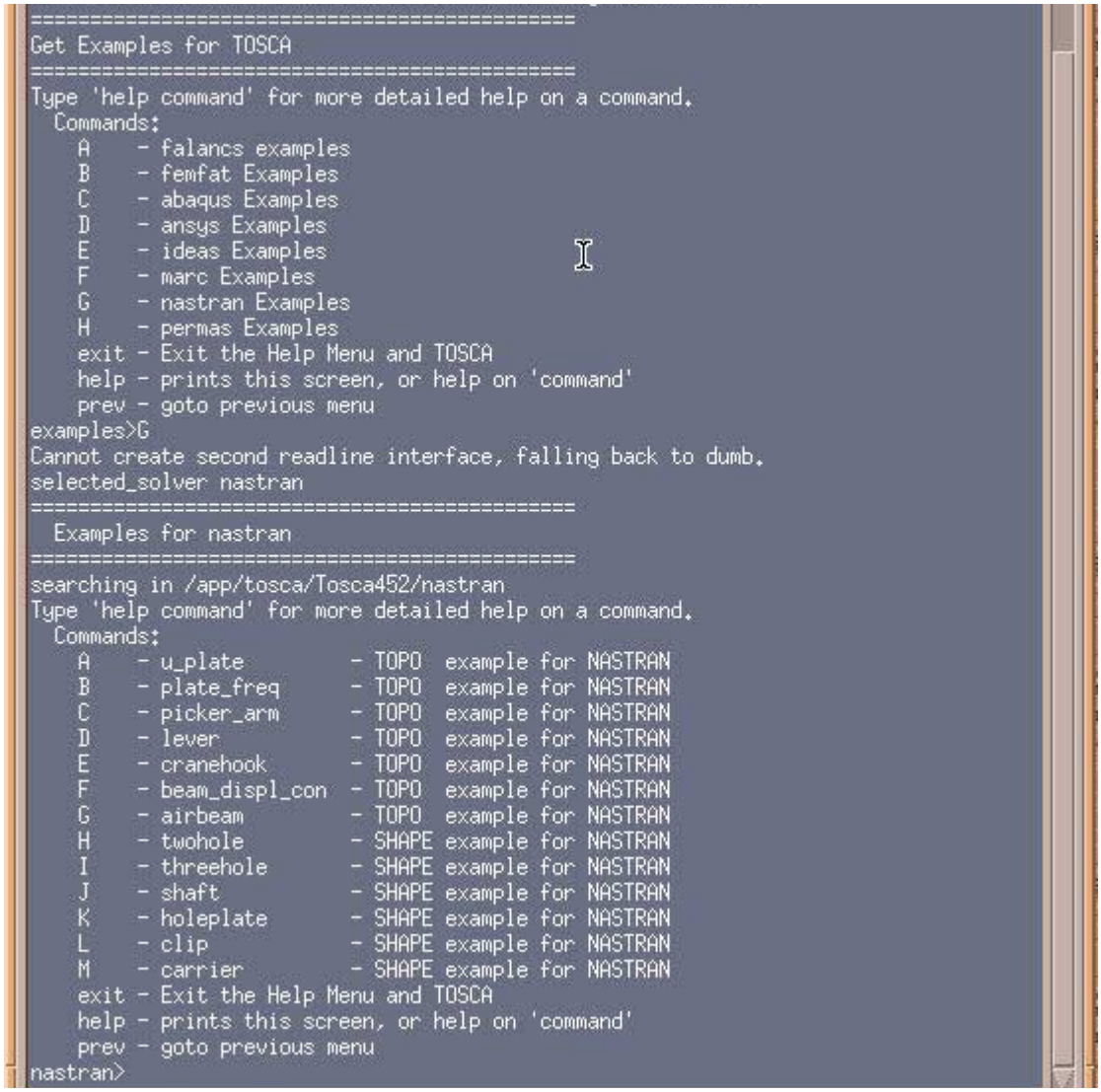
commandline call:

```
tosca50 -help
```

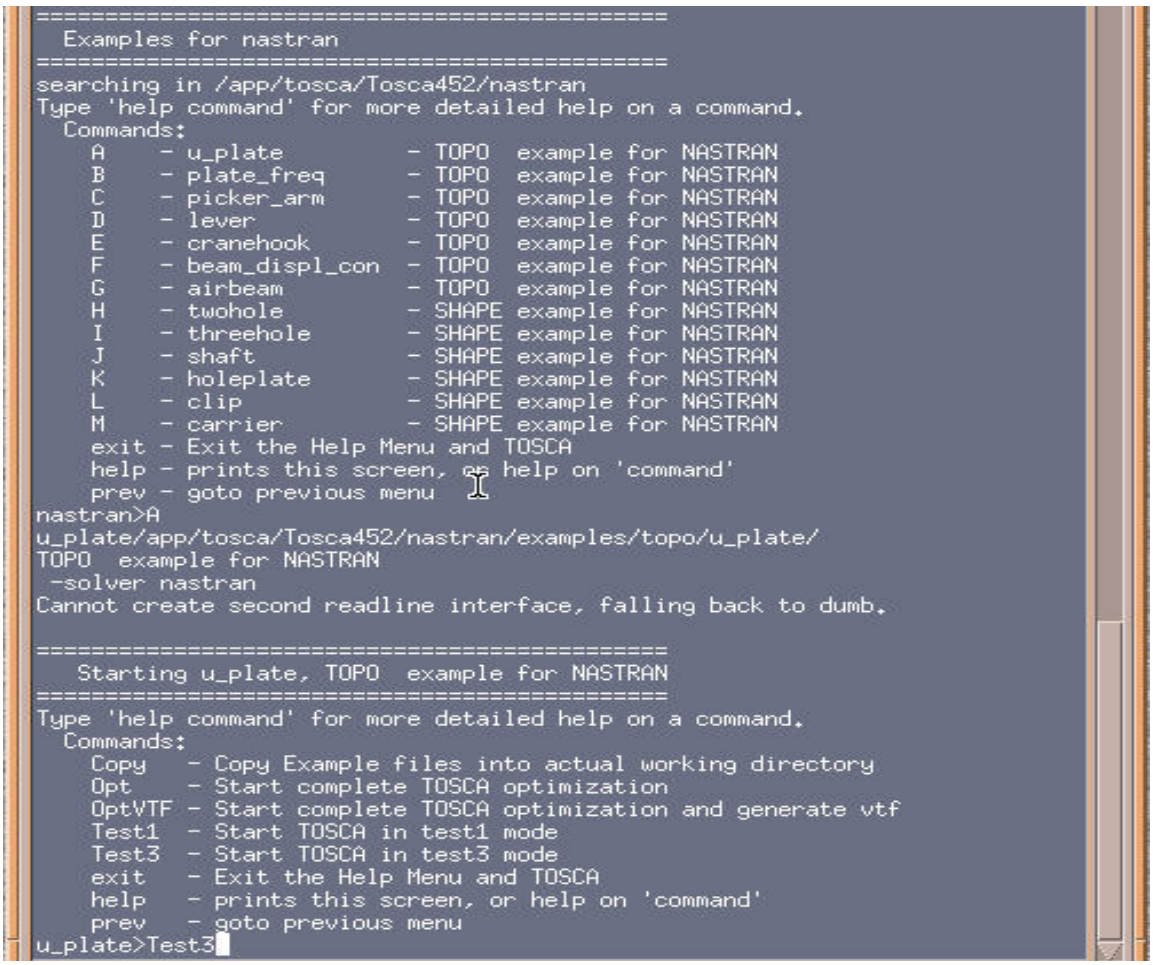
Select D for getting an example

fed-hp2 114: cd /scratch/
fed-hp2 115: mkdir tosca_test
fed-hp2 116: cd tosca_test/
fed-hp2 117: tosca452 -help
OPTIMIZATION CONTROL FE Design TOSCA
TOSCA.Topo and TOSCA.Shape are registered trademarks
and service marks of the FE Design GmbH
Copyright(c) May 2002 by FE-DESIGN GmbH, all rights reserved
PERL execution control
Version 4.5.2
FE-DESIGN GmbH, Karlsruhe, Germany
http://www.fe-design.de
Build: 15.08.2003
TOSCA Time information 11.09.2003 at 19:13:01
Optimization runs on a hpux system
READING TOSCA CONFIGURATION
Reading TOSCA Configuration file : /app/tosca/Tosca452/bin/tosca_ctrl.cfg
... Reading Tosca 452 configuration
TOSCA - Online Help
Type 'help command' for more detailed help on a command.
Commands:
A - Commandline options
B - TOSCA configuration
C - TOSCA File Save settings
D - Copy and Run Examples for TOSCA
exit - Exit the Help Menu and TOSCA
help - prints this screen, or help on 'command'
return - Leave HELP Menu and Return to TOSCA job
tosca_help>

Select the finite element solver


==========
Get Examples for TOSCA
==========
Type 'help command' for more detailed help on a command.
Commands:
A - falancs examples
B - femfat Examples
C - abaqus Examples
D - ansys Examples
E - ideas Examples
F - marc Examples
G - nastran Examples
H - permas Examples
exit - Exit the Help Menu and TOSCA
help - prints this screen, or help on 'command'
prev - goto previous menu
examples>G
Cannot create second readline interface, falling back to dumb.
selected_solver nastran
==========
Examples for nastran
==========
searching in /app/tosca/Tosca452/nastran
Type 'help command' for more detailed help on a command.
Commands:
A - u_plate - TOPO example for NASTRAN
B - plate_freq - TOPO example for NASTRAN
C - picker_arm - TOPO example for NASTRAN
D - lever - TOPO example for NASTRAN
E - cranehook - TOPO example for NASTRAN
F - beam_displ_con - TOPO example for NASTRAN
G - airbeam - TOPO example for NASTRAN
H - twohole - SHAPE example for NASTRAN
I - threehole - SHAPE example for NASTRAN
J - shaft - SHAPE example for NASTRAN
K - holeplate - SHAPE example for NASTRAN
L - clip - SHAPE example for NASTRAN
M - carrier - SHAPE example for NASTRAN
exit - Exit the Help Menu and TOSCA
help - prints this screen, or help on 'command'
prev - goto previous menu
nastran>

Select an example for topology, shape or bead optimization. The detailed description of the examples may be found in the Examples Manual of the TOSCA installation.


==
Examples for nastran
==
searching in /app/tosca/Tosca452/nastran
Type 'help command' for more detailed help on a command.
Commands:
A - u_plate - TOPO example for NASTRAN
B - plate_freq - TOPO example for NASTRAN
C - picker_arm - TOPO example for NASTRAN
D - lever - TOPO example for NASTRAN
E - cranehook - TOPO example for NASTRAN
F - beam_displ_con - TOPO example for NASTRAN
G - airbeam - TOPO example for NASTRAN
H - twohole - SHAPE example for NASTRAN
I - threehole - SHAPE example for NASTRAN
J - shaft - SHAPE example for NASTRAN
K - holeplate - SHAPE example for NASTRAN
L - clip - SHAPE example for NASTRAN
M - carrier - SHAPE example for NASTRAN
exit - Exit the Help Menu and TOSCA
help - prints this screen, or help on 'command'
prev - goto previous menu
nastran>A
u_plate/app/tosca/Tosca452/nastran/examples/topo/u_plate/
TOPO example for NASTRAN
-solver nastran
Cannot create second readline interface, falling back to dumb.
==
Starting u_plate, TOPO example for NASTRAN
==
Type 'help command' for more detailed help on a command.
Commands:
Copy - Copy Example files into actual working directory
Opt - Start complete TOSCA optimization
OptVTF - Start complete TOSCA optimization and generate vtf
Test1 - Start TOSCA in test1 mode
Test3 - Start TOSCA in test3 mode
exit - Exit the Help Menu and TOSCA
help - prints this screen, or help on 'command'
prev - goto previous menu
u_plate>Test3

First the model is selected (in this case ’A’ for the u_plate model). In the next dialog, ’Test3’ or ’Opt’ is selected to start the optimization procedure directly with TOSCA.

processing /app/tosca/Tosca452/nastran/examples/topo/u_plate/u_plate.bdf
processing /app/tosca/Tosca452/nastran/examples/topo/u_plate/u_plate.par
starting TOSCA ...
/app/tosca/Tosca452/bin/os_hpux/tosca_perl.e u_plate -solver nastran -type test3
OPTIMIZATION CONTROL FE Design TOSCA
TOSCA.Topo and TOSCA.Shape are registered trademarks
and service marks of the FE Design GmbH
Copyright(c) May 2002 by FE-DESIGN GmbH, all rights reserved
PERL execution control
Version 4.5.2
FE-DESIGN GmbH, Karlsruhe, Germany
http://www.fe-design.de
Build: 15.08.2003
TOSCA Time information 11.09.2003 at 19:14:54
Optimization runs on a hpux system
READING TOSCA CONFIGURATION
Reading TOSCA Configuration file : /app/tosca/Tosca452/bin/tosca_ctrl.cfg
... Reading Tosca 452 configuration
Actual settings for optimization job :
actual working dir : /scratch/tosca_test
environment TOSCA : /app/tosca/Tosca452
environment HOME : /users/brandt
jobname : u_plate
fe-input File : u_plate.bdf
parameter file : u_plate.par
fe_solver : nastran
fe_solver_exe : "/usr/bin/nastran"
fe_solver_call : "/usr/bin/nastran" u_plate.bdf batch=no notify=no
opt_type : test3
Prep-module: "/app/tosca/Tosca452/bin/tosca_prep"
Opt -module: "/app/tosca/Tosca452/bin/tosca_opt"
Act Directory : /scratch/tosca_test
jobname : u_plate

After succesful completion of the optimization job, the user is returned to the help menu. In the actual working directory you will find the input files and the jobname result directory.

8 Appendix

8.1 File structure in the installation directory

8.1.1 Directories and files in the TOSCA installation folder

abaqus	**examples**	**shape**	**carrier** **carrier_stamp** **clip** **holeplate** **shaft** **shaft_drill** **shaft_turn** **threehole** **twohole**	jobname.inp jobname.par	Examples: Analysis model and parameter file
		topo	**airbeam** **airbeam_sym** **beam_displ_con** **cranehook** **lever** **pickerarm** **plate_freq** **palte_min_max** **plate_sysm_freq** **steering_arm** **u_plate**	jobname.inp jobname.par	Examples: Analysis model and parameter file
		bead	**plate**	jobname.inp jobname.par	Examples: Analysis model and parameter file
	command.ini				configuration file

Tabelle 1 Structure of the TOSCA installation folder

Version 5.0 Rev. 01 - 03.2004

ansys	**examples**	**shape**	**carrier** **carrier_stamp** **clip** **holeplate** **shaft** **shaft_drill** **shaft_turn** **threehole** **twohole**	jobname.ans jobname.par	Examples: Analysis model and parameter file
		topo	**airbeam** **airbeam_sym** **beam_displ_con** **cranehook** **lever** **pickerarm** **plate_freq** **palte_min_max** **plate_sysm_freq** **steering_arm** **u_plate**	jobname.ans jobname.par	Examples: Analysis model and parameter file
		bead	**plate**	jobname.inp jobname.par	Examples: Analysis model and parameter file
	command.ini				configuration file
bin	**os_aix** **os_hpux** **os_irix** **os_linux** **os_solaris** **os_winnt**				platform specific executables
	tosca(.cmd) etc. tosca_ctrl.cfg etc.				scripts and configuration files

Tabelle 1 Structure of the TOSCA installation folder

docu	tosca_v50_de.pdf			TOSCA documentation (german)
	tosca_v50_en.pdf			TOSCA documentation (english)
	release_notes_v50_de.pdf			Releasenotes TOSCA50 (german)
	release_notes_v50_en.pdf			Releasenotes TOSCA50 (english)
	install_tosca50_de.pdf			Installation Guid (german)
	install_tosca50_en.pdf			Installation Guide (english)
falancs	**examples**	**abaqus** **ansys** **ideas** **nastran**	holeplate_dam.par holeplate_dam.fal holeplate_dam.ext time_his.txt	Examples: Analysis model and parameter file
	tosca_falancs.ini			configuration file
femfat	examples	abaqus ansys ideas nastran	holeplate_dam.par holeplate_dam.ffj holeplate_dam.ext st37.ffd	Examples: Analysis model and parameter file

Tabelle 1 Structure of the TOSCA installation folder

ideas	**examples**	**shape**	**carrier** **carrier_stamp** **clip** **holeplate** **shaft** **shaft_drill** **shaft_turn** **threehole** **twohole**	jobname.sun jobname.unv jobname.par	Examples: Analysis model and parameter file
		topo	**airbeam** **airbeam_sym** **beam_displ_con** **cranehook** **lever** **pickerarm** **plate_freq** **palte_min_max** **plate_sysm_freq** **steering_arm** **u_plate**	jobname.sun jobname.unv jobname.par	Examples: Analysis model and parameter file
		bead	**plate**	jobname.sun jobname.unv jobname.par	Examples: Analysis model and parameter file
	command.ini				configuration file
java	**lib_java**	TOSCApre.jar toscagui.jar			TOSCA Java components
	os_aix **os_hpux** **os_irix** **os_linux** **os_solaris** **os_winnt**	jre			platform specific java runtime environments
macros	stop1.mac etc.				TOSCA macros

Tabelle 1 Structure of the TOSCA installation folder

marc	**examples**	**shape**	**carrier** **carrier_stamp** **clip** **holeplate** **shaft** **shaft_drill** **shaft_turn** **threehole** **twohole**	jobname.dat jobname.par	Examples: Analysis model and parameter file
		topo	**airbeam** **airbeam_sym** **beam_displ_con** **cranehook** **lever** **pickerarm** **plate_freq** **palte_min_max** **plate_sysm_freq** **steering_arm** **u_plate**	jobname.dat jobname.par	Examples: Analysis model and parameter file
		bead	**plate**	jobname.dat jobname.par	Examples: Analysis model and parameter file
	command.ini				configuration file

Tabelle 1 Structure of the TOSCA installation folder

nastran	**examples**	**shape**	**carrier** **carrier_stamp** **clip** **holeplate** **shaft** **shaft_drill** **shaft_turn** **threehole** **twohole**	jobname.bdf jobname.par	Examples: Analysis model and parameter file
		topo	**airbeam** **airbeam_sym** **beam_displ_con** **cranehook** **lever** **pickerarm** **plate_freq** **palte_min_max** **plate_sysm_freq** **steering_arm** **u_plate**	jobname.bdf jobname.par	Examples: Analysis model and parameter file
		bead	**plate**	jobname.bdf jobname.par	Examples: Analysis model and parameter file
	macros	Makros for MSC.Nastran			Marcos for the simultaneous generation of xdb and op2 result files.
	command.ini				configuration file

Tabelle 1 Structure of the TOSCA installation folder

Version 5.0 Rev. 01 - 03.2004

permas	**examples**	**shape**	**carrier** **carrier_stamp** **clip** **holeplate** **shaft** **shaft_drill** **shaft_turn** **threehole** **twohole**	jobname.dat jobname.uci jobname.par	Examples: Analysis model and parameter file
		topo	**airbeam** **airbeam_sym** **beam_displ_con** **cranehook** **lever** **pickerarm** **plate_freq** **palte_min_max** **plate_sysm_freq** **steering_arm** **u_plate**	jobname.dat jobname.par jobname.uci	Examples: Analysis model and parameter file
		bead	**plate**	jobname.dat jobname.par jobname.uci	Examples: Analysis model and parameter file
	command.ini				configuration file
UninstallerData					Uninstaller information
Tosca_5.0_InstallLog.xml					INstallation protocol file

Tabelle 1 Structure of the TOSCA installation folder

8.1.2 Files in the $tosca/bin directory

The subdirectory ’bin’ in the TOSCA installation directory ($tosca) contains the startscripts for the different TOSCA modules and processes. The executable files are located in the corresponding platform directories. The necessary environment variables are set in the startscripts and the executables are loaded.

$tosca/bin contains the following scripts:

common_env		common settings for the startscripts e.g. environment variables and path enhancements. The script is used in all startscripts.
tosca_ctrl.cfg		Configuration file of the TOSCA control program
tosca_post.cfg		Configuration file for the tosca_post module
InstallToscaLinks		Setting the symbolic links resp. the entries in the Windows program's menu
fem_include		Startscript for os_<platform>/fem_include
fem2onf		Startscript for os_<platform>/fem2onf
fem_modif		Startscript for os_<platform>/fem_modif
life2onf		Startscript for os_<platform>/life2onf
onf2bof		Startscript for os_<platform>/onf2bof
onf2odb		Startscript for os_<platform>/onf2odb
onf2rst		Startscript for os_<platform>/onf2rst
onf2uns		Startscript for os_<platform>/onf2uns
onf2vtf		Startscript for os_<platform>/onf2vtf
res2onf		Startscript for os_<platform>/res2onf
tosca		Startscript for os_<platform>/tosca
tosca5_script		Reference for the /usr/bin links
tosca_opt		Startscript for os_<platform>/tosca_opt
tosca_perl		Startscript for tosca_perl
tosca_post		Startscript for tosca_post
tosca_prep		Startscript for os_<platform>/tosca_prep
tosca_shell		Reference for the TOSCA shell in the Windows program's menu
tosca_smooth		Startscript for os_<platform>/tosca_smooth
tosca_view		Startscript for os_<platform>/tosca_view

Tabelle 2 Files in $tosca/bin

These scripts exist for Windows (.cmd extension) and Unix. They load the native executables on the corresponding platform. On Unix systems the library paths are also extended in these scripts.

$tosca/bin/os_xxx (xxx=platformname)

contains the following executable files (on Unix systems without the .exe extension)

Executable	function
tosca_server.exe	Checks the licensing and starts the optimization
tosca_perl.exe	TOSCA optimization control program
fem_include.exe	Determination of the include files in the finite element model
tosca_prep.exe	TOSCA optimization preprocessing
fem2onf.exe	Finite element input to ONF translator
fem_modif.exe	Modification of the finite element model
res2onf.exe	Converter for the finite element resultfiles
tosca_opt.exe	TOSCA optimization module
life2onf.exe	Conversion of durability results
tosca_smooth.exe	TOSCA.smooth
tosca_post.exe	TOSCA.post
onf2odb.exe	Conversion in Abaqus CAE Format
onf2rst.exe	Conversion in Ansys Format
onf2bof.exe	Conversion in Medina Format
onf2unv.exe	Conversion in I-Deas Format
onf2vtf.exe	Conversion in vtf Format
tosca_view	TOSCA vtf Viewer
tosca_gui.exe	TOSCA.gui Starter
tosca_gui.lax	Definition of the Preferences for TOSCA.gui
tosca_client	TOSCA license utility
lmutil	Flexlm utilities

Tabelle 3 Programs in the os_<platform> directories (here: Windows)

8.2 Configuration files

8.2.1 tosca_ctrl.cfg

The configuration file of the TOSCA optimization control program is completely implemented in PERL syntax and is interpreted at the beginning of the optimization procedure. All variables that are embedded in ampersands (&VAR&) are replaced during the installation procedure. Single parameters are documented in the Commands Manual of the TOSCA documentation.

Example: MSC.Nastran Configuration:

```
###### NASTRAN ######
```

During the start of optimization, the fe_solver variable is defined. In the configuration the different solver configuration settings are enclosed in if - else - endif statements.

```
if ($fe_solver} eq "nastran") {
The $tosca_interface variable defines the interface
  $ENV{"tosca_interface"} = "${tosca}/nastran";
```

As path names are represented differently in Windows and Unix, TOSCA determines on which platform the optimization has been started. The $Client Object is filled with the actual platform information at the beginning of the TOSCA run and can be used for similar conditional statements. The following conditions are possible:

$Client->isUnix

$Client->isAix

$Client->isHpux

$Client->isIrix

$Client->isLinux

$Client->isSolaris

```
if ($Client->isWin) {
```

The TOSCA variable $fe_solver_exe is set to the full path name of Nastran. This value is defined during the installation and TOSCA verifies the existence of this executable or startscript at the beginning.

e.g. for Windows systems : C.\Program files\msc\msc2001\i386\nastran.exe

```
 ${fe_solver_exe}="&PATH_OF_SELECTED_NASTRAN_WIN&&USER_SELECTED_NASTRAN_WIN&";
```

Additional Nastran command line parameters are set to the $fe_solver_call variable. These parameters are used for the start of Nastran within TOSCA.

```
 ${fe_solver_call} = "${fe_solver_exe} ${modelname}.${inp_ext} notify=no";
} else {
```

In this conditional block the equivalent settings for Unix clients are defined.

```
 ${fe_solver_exe}="&PATH_OF_SELECTED_NASTRAN_UX&&USER_SELECTED_NASTRAN_UX&";
 ${fe_solver_call} = "${fe_solver_exe} ${modelname}.${inp_ext} batch=no notify=no";
}
```

Common information is set outside the platform specific blocks.

```
${res_ext} = "xdb";
${solver_add_call} = "";
```

...

The next section shows the installation template of this configuration file. The installation routine enters the information pertaining to the finite element solver installation paths, the required solver interfaces etc. that are provided with the installation dialogs.

```
# Configuration File for TOSCA V5.0

print "... Reading TOSCA 5.0 Configuration";

# ~~~
```

```
${fem_modif} = "${tosca}/bin/fem_modif";
${fem_modif_add_call} ="";

${fem2onf}   = "${tosca}/bin/fem2onf";
$ENV{fem2onf} = ${fem2onf};

${res2onf}   = "${tosca}/bin/res2onf";
${res2onf_add_call} ="-bin";

${fem_include}  = "${tosca}/bin/fem_include";

${life2onf}  = "${tosca}/bin/life2onf";

${ask_before_delete} = "no";

set_include_search_path("${tosca}/macros",
                        "${tosca}/nastran/macros");

# ~~~
```

```
  } elsif (${fe_solver} eq "abaqus") {
    $ENV{"tosca_interface"} = "${tosca}/abaqus";
    if ($Client->isWin) {
      ${fe_solver_exe} = "&IA_PATH_ABAQUS_WIN&";
    }else{
      ${fe_solver_exe} = "&IA_PATH_ABAQUS_UX&";
    }
    ${fe_solver_call} = "${fe_solver_exe} job=${modelname} analysis interactive";

    ${res_ext} = "fil";
    ${fe_solver_add_call} = "";

    add_move_per_iter_list("first_last", "SAVE.${res_ext}", "*.${res_ext}");
    add_move_per_iter_list("first_last", "SAVE.odb", "*.odb");
    add_move_per_iter_list("first_last", "SAVE.msg", "*.msg");
    add_move_per_iter_list("first_last", "SAVE.dat", "*.dat");
    add_move_per_iter_list("ever", "SAVE.${inp_ext}", "*.${inp_ext}");


    add_delete_after_opt_list("*.com", "*.dat", "*.prt", "*.sta", "*.msg");
    set_copy_solver_info_list("ever", "SAVE.${inp_ext}", "*.${inp_ext}" );

# ~~~
```

```
# ~~~
```

```
    add_move_per_iter_list("ever", "SAVE.${inp_ext}", "${modelname}.uci");

    add_delete_after_opt_list("*.pro*", "*.res*");
    set_copy_solver_info_list("ever", "SAVE.dat", "${modelname}.dat");
    add_copy_solver_info_list("ever", "SAVE.dat", "${modelname}.uci");

# ~~~
```

```
# ~~~
```

```
    if (${fe_solver} eq "nastran"){
      add_move_per_iter_list("first_last", "SAVE.txt", "${modelname}.txt" );
      add_move_per_iter_list("first_last", "SAVE.dam", "${modelname}.dam" );
      add_move_per_iter_list("first_last", "SAVE.HS", "${modelname}_HS.txt");
      add_move_per_iter_list("first_last", "SAVE.HS", "${modelname}_HS.tsi" );
      add_move_per_iter_list("first_last", "SAVE.op2", "${modelname}.op2" );

    }elsif (${fe_solver} eq "abaqus") {
      add_move_per_iter_list("first_last", "SAVE.txt", "${modelname}.txt" );
      add_move_per_iter_list("first_last", "SAVE.fdm", "${modelname}.fdm"  );
      add_move_per_iter_list("first_last", "SAVE.HS", "${modelname}_HS.txt");
      add_move_per_iter_list("first_last", "SAVE.HS", "${modelname}_HS.tsi" );

    }elsif (${fe_solver} eq "ansys") {
      add_move_per_iter_list("first_last", "SAVE.txt", "${modelname}.txt");
      add_move_per_iter_list("first_last", "SAVE.HS", "${modelname}_HS.txt" );
      add_move_per_iter_list("first_last", "SAVE.HS", "${modelname}_HS.tsi");

    }elsif ($fe_solver eq "ideas") {
      add_move_per_iter_list("first_last", "SAVE.txt", "${modelname}.txt" );
      add_move_per_iter_list("first_last", "SAVE.udm", "${modelname}.udm"   );
      add_move_per_iter_list("first_last", "SAVE.HS", "${modelname}_HS.txt");
      add_move_per_iter_list("first_last", "SAVE.HS", "${modelname}_HS.tsi" );
    };

# ~~~
```

```
 }
# ~~~
```

```
###### Global File Save Information ######
 add_move_per_iter_xxx_list("ever", "SAVE.onf", "SHAPE_???.onf");
 add_move_per_iter_xxx_list("ever", "SAVE.onf", "TP_???.onf");
 add_move_per_iter_xxx_list("ever", "SAVE.onf", "BEAD_???.onf");
 add_move_per_iter_xxx_list("ever", "SAVE.onf", "MODEL_???.onf");
 add_move_per_iter_xxx_list("ever", "SAVE.onf", "MODEL_fem.onf");
 add_move_per_iter_xxx_list("ever", "SAVE.vtf", "*_???_*.vtf");
 add_move_per_iter_xxx_list("ever", "SAVE.wrl", "*_???.wrl");
 add_move_per_iter_xxx_list("ever", "SAVE.wrl", "*_???.WRL");


###### Move files after optimization ######
        add_move_after_opt_list("TOSCA_POST",    "*.els.*",    "*.nod.*",    "*.grp*",    "*.out_iso*",
"*.out_fem*","TOSCA.OUT", "*.txt", "*.neu");
 add_move_after_opt_list("SAVE.wrl", "*.WRL", "*.wrl");
 add_move_after_opt_list("SAVE.html", "*.html", "*.HTML");
 add_move_after_opt_list("TOSCA_DB", "TOSCA_DB.*", "SYS_DB.LIM", "TOSCA.LOG");
   add_move_after_opt_list("SAVE.onf",  "*SHAPE*.onf",  "*BEAD*.onf",  "TP*.onf",  "GROUPS.onf",
"TP_GROUPS.onf", "*_fem.onf", "MODEL_*.onf");
 add_move_after_opt_list("SAVE.stl", "*.stl", "*.STL");

###### optimization database ######
     set_opt_db_list("SYS_DB.LIM",   "TOSCA_DB.FIM",   "TOSCA_DB.FIX",   "TOSCA_DB.V1M",
"TOSCA_DB.V1X", "TOSCA_DB.V2M", "TOSCA_DB.V2X");


###### file delete info  ######
   add_delete_after_opt_list("*.log*",  "*.tmp",  "*.mac",  "*.onf",  "tosca.restart",  "STOP_DAT.CAO",
"*.par_cfg" );
```

8.2.2 tosca_post.cfg

The tosca_post.cfg file contains the configuration for the tosca_post commandline module. This configuration file is also completely implemented in PERL syntax. The different onf2post modules are configured and the path to the MSC.Patran executable file is specified.

```
$opt_sys = "TOSCA";
$tosca_bin="$ENV{tosca}/bin";


if ($client->isWin) {
                $onf2rst="$tosca_bin/onf2rst.cmd";
                $onf2odb="$tosca_bin/onf2odb.cmd";
                $onf2bof="$tosca_bin/onf2bof.cmd";
                $onf2unv="$tosca_bin/onf2unv.cmd";
```

```
                $onf2vtf="$tosca_bin/onf2vtf.cmd";

                $MedinaDscatPath="${tosca_bin}/os_winnt/dscat.ds";
                $PatranCall="&IA_PATH_PATRAN_WIN&";

        } else {

                $PatranCall="&IA_PATH_PATRAN_UX&";

                $onf2rst="$tosca_bin/onf2rst";
                $onf2odb="$tosca_bin/onf2odb";
                $onf2bof="$tosca_bin/onf2bof";
                $onf2unv="$tosca_bin/onf2unv";
                $onf2vtf="$tosca_bin/onf2vtf";
                if ($client->isAix) {
                        $MedinaDscatPath="${tosca_bin}/os_aix/dscat.ds";
                } elsif ($client->isHpux) {
                        $MedinaDscatPath="${tosca_bin}/os_hpux/dscat.ds";
                } elsif ($client->isIrix) {
                        $MedinaDscatPath="${tosca_bin}/os_irix/dscat.ds";
                } elsif ($client->isLinux) {
                        $MedinaDscatPath="${tosca_bin}/os_linux/dscat.ds";
                } elsif ($client->isSolaris) {
                        $MedinaDscatPath="${tosca_bin}/os_solaris/dscat.ds";
                } else {
                        print "Unkown Unix platform\n";
                        print "Please check if you try to run tosca_post on an unsupported platform\n";
                        exit(0);
                }
        }
```

8.2.3 Startscripts

The next section demonstrates the use and structure of the startscript in $tosca/bin for Windows and Unix.

8.2.3.1 Unix

Bourne shell syntax is used for the startscripts on Unix systems. This ensures maximum compatibility among the supported platforms. During installation, the name of the TOSCA installation directory replaces the &TOSCA_INSTALL_DIR& Variable.

A uname call is used to obtain information regarding the actual operating system.

If any runtime libraries are provided, the library path is extended.

If any problems concerning the runtime libraries occur, the settings may be checked and corrected in the startup scripts.

The /usr/bin/tosca50 link links to the tosca5_script in the $tosca/bin directory. This central startup script is described in more detail below:

```
#!/bin/sh
```

This script first sets the environment variable <tosca> to the TOSCA installation directory. This variable is used by all TOSCA components to determine the installation path and search for configuration files, examples etc.

```
tosca=&TOSCA_INSTALL_DIR&
export tosca
```

The next step is the extension of the search path:

```
PATH=$tosca/bin:$PATH
export PATH
```

The online documentation may be started using this script. The path to the acrobat reader must be specified correctly

```
function docustart {
echo "
"
echo "-------------------------------------------------------------------------------"
echo "  tosca started online documentation"
echo "-------------------------------------------------------------------------------"
 &USER_SELECTED_ACROREAD& ${tosca}/docu/&TOSCA_DOCU&.pdf &
}
```

In case the TOSCA batch call (tosca50) has been chosen, TOSCA can be set to execute in the background by adding the command line option : -nohup.

```
if [ $runmode = "nohup" ]; then
   printstart
   (nohup &TOSCA_INSTALL_DIR&/bin/tosca $arglist > $jobname.log ; printstop )&
else
   &TOSCA_INSTALL_DIR&/bin/tosca  $arglist
fi
exit
```

8.2.3.2 Windows

The structure of the Windows startscripts is similar.

```
@echo off
set tosca=&TOSCA_INSTALL_DIR&
set PATH=%tosca%\bin;%PATH%

&TOSCA_INSTALL_DIR&\bin\OS_WINNT\tosca_server.exe %*
```

On Windows systems a link exists in the program's menu that starts the following script:

```
@echo off
set tosca=&TOSCA_INSTALL_DIR&
set PATH=&TOSCA_INSTALL_DIR&\bin;%PATH%

start
```

A command shell is opened that contains all the necessary variables set in the environment.

8.3 Troubleshooting

The following chapters explain possible errors that can occur while starting TOSCA

8.3.1 Configuration

8.3.1.1 tosca50 or other tosca modules cannot be found

Please check if a client Installation has been performed on your computer. By default the TOSCA links are generated in the /usr/bin on UNIX sytems and in the program's menu on Windows systems.

In case the client installation was not performed for your computer, you can call the installation scripts InstallToscaClient.sh or InstallToscaClient.cmd placed in the <tosca_install_dir>/bin directory. On a Unix system this script will generate symbolic links in /usr/bin, on a windows system there will be generated links in the programs menu.

Please check if there is sufficient authorisation for the symbolic links for all users. If this is not the case, authorisation may be adapted using e.g

```
chmod -R 755 <Tosca50>
```

8.3.1.2 Finite element solver executables not defined correctly

During the installation process, the paths to the fe solver executables are requested and entered to the configuration file: <tosca_install_dir>/bin/tosca_ctrl.cfg.

In case these entries are empty or incorrect the following error message will appear:

```
Reading TOSCA Configuration file : /scratch/test/Tosca50/bin/tosca_ctrl.cfg
... Reading TOSCA 5.0 Configuration
 ------------------------------------------------------------------------
|               TOSCA COMMAND_SHELL ERROR
 ------------------------------------------------------------------------
Wrong configuration settings. The following configuration
 variable is not set correctly:

 Variable_name      : $fe_solver_exe
 Variable_value     :

Path to fe-solver executable is not set correctly
 ------------------------------------------------------------------------
Configuration is not set correctly
```

To fix this problem you need to enter the correct pathes to the fe solver executables either in the main configuration file or an user specific configuration file in the users HOME directory.

8.3.1.3 Path to runtime libraries is not correct

TOSCA depends on several runtime libraries. As far as they are not part of the operating system these libraries will be copied to the TOSCA installation directory. The installation directory is added to the runtime library search paths for the TOSCA processes. The runtime library search paths are defined in the scripts common_env.sh and common_env.cmd. These scripts reside in <tosca_install_dir>/bin.

In case the runtime libraries cannot not be found there will appear an error message similar to the following:

```
/usr/lib/pa20_64/dld.sl: Unable to find library 'libHKSbdo-A.sl'.
/scratch/test/Tosca50/bin/fem2onf[29]: 5284 Memory fault(coredump)
```

8.3.1.4 /tmp directory is full

The TOSCA optimization process is controlled by the component tosca_perl. This is a perl application, which has been packaged to a self extracting executable. At program start this executable is extracted to a temporary file system. This requires approximately two megabyte free disk space in the /tmp file system. In case the actual user has no write privileges for /tmp or the /tmp file system has not enough free space the following error message is generated:

```
fed-hp2 128: tosca50 clip -solver abaqus
mkdir("/tmp/pdk-brandt/") failed, errno=13
mkdir("/tmp/pdk-brandt/") failed, errno=13
Can't find temp directory at perlapp line 1063.
BEGIN failed--compilation aborted at /PerlApp/Term/Shell.pm line 5.
BEGIN failed--compilation aborted at tosca_perl.pl line 8.
```

This problem can be fixed by either providing enough space in /tmp or setting the environment variable PAR_TEMP to a file system with aproriate free space. This variable can be defined in the comm_env.sh scripts to be available for each user.

8.3.1.5 java virtual machines are not accessible

The TOSCA graphical user interface tosca_gui requires a java virtual machine. This is either installed with TOSCA or already existing on the client computer. The installation directory of the java virtual machines is defined in the configuration scripts: <tosca_install_dir>/bin/os_>platform>/todsca_gui.lax

Error message:

```
fed-hp2 129: tosca50_gui
No Java virtual machine could be found from your PATH
environment variable.  You must install a VM prior to
running this program.
```

related settings in tosca_gui.lax

```
#   LAX.NL.CURRENT.VM
#   ----------------
#   the VM to use for the next launch

lax.nl.current.vm=/opt/Tosca50/java/os_hpux/jre
```

8.3.2 Licensing

Problems with the FlexLM authorisation may occur in two stages of the optimization process-

If the license server cannot be accessed or the host ID is not correct, TOSCA will stop at the beginning of the optimization run. The corresponding error messages are printed out to the screen and to the Auth.log file in the start directory of the optimization job.

```
fed-hp2 134: tosca50 clip -solver abaqus
Authorisation system reports fatal FLEXLM Error:
Error description: Server node is down or not responding
 See the system adminstrator about starting the server, or
 make sure the you're referring to the right host (see LM_LICENSE_FILE)
Feature:       TOSCA_CORE
Hostname:      fed-gulli1
License path:  7777@fed-gulli1
FLEXlm error:  -96,7

For further information, refer to the FLEXlm End User Manual,
available at "www.macrovision.com".
-------------------------------------------------------------------------
Tosca   ERROR
Program will finish now.
```

If problems occur concerning the licensed TOSCA funtions (e.g. the solver interface is not licensed correctly), TOSCA will be aborted during the optimization procedure. In case of a non-existing interface license, TOSCA will be aborted in the optimization preprocessing module. The corresponding error messages may be found on the screen and the TOSCA.OUT file in the <jobname>/DB_ERROR directory.

```
 -------------------------------------------------------------------------
 NAME: /scratch/test/Tosca50/abaqus/command.ini

#INPUT: FE_PIPE, ALL, ABAQUS
 ******** TOSCA WARNING ********
 Interface : ABAQUS not authorised
 Hint :
   LIST, AUTHORISATION -->> see what is allowed
#INPUT: OPTIONS
#INPUT:   READ_TOPO_RESULTS = NODE
#INPUT:   READ_LOAD         = ON
#INPUT: END_
 -------------------------------------------------------------------------
```

The available interfaces may be listed by the LIST, AUTH command

Version 5.0 Rev. 01 - 03.2004

```
| AUTHORISATION INFORMATION of TOSCA                                    |
 -----------------------------------------------------------------------
| NAME                                                  STATUS    |
| Description                                                     |
 -----------------------------------------------------------------------
 Available features and interfaces:
 -----------------------------------------------------------------------
|TOSCA_CORE                                                      X |
|Main Feature of authorisation process                             |
 -----------------------------------------------------------------------
|TOSCA_INT_NASTRAN                                                 X |
|Feature for NASTRAN interface                                     |
 -----------------------------------------------------------------------
|TOSCA_INT_ANSYS                                                   X |
|Feature for ANSYS interface                                       |
 -----------------------------------------------------------------------
```

In addition to the error messages in the TOSCA protocol files, different tools are available to find the cause of the error. On Windows systems the lmtools utility may be used to get detailed information pertaining to the status of the license server and the licenses available.

8.3.2.1 Checklist

- Does a valid license file exist ?
  The default installation places the license.dat file in the $tosca/license directory. The header must contain the servername, the serverport and the path to the FED vendor daemon. The host ID must be the same as the host ID of your license server.
- Is the license server started ?
  Windows: The Task Manager can be used to check if the lmgrd process is running.
  Unix: ps -eadf|grep lmgrd
- Is the FED vendor daemon started ?
  The name of the vendor daemon process is FED - it must be started on the license server. This can be checked the same way as the lmgrd process.
- Is the license path correct ?
  FlexLM searches the license server path in a license server list . This list is made up of information from the environment of the current user:
  - Information in $HOME/.flexlmrc or HKLM\SOFTWARE\Flexlm License manager

- environment variable LM_LICENSE_FILE
- Environment variable FED_LICENSE_FILE
  Definition of the environment variable FED_LICENSE_FILE in the file <tosca_install_dir>/bin/common_env

  the valid license server path may be determined using the <lmutil lmpath -status> utility.

- Can the TOSCA license be checked out ?
  The tosca_client program is the easiest way for the user to check the TOSCA_CORE license.

8.3.2.2 TOSCA Test utilities

- tosca_client
  command:
  tosca_client [License Name] [License Version] [-c License Path]
  Example:
  tosca_client TOSCA_CORE 5.0 -c /opt/Tosca50/license/license.dat

  tosca_client can check out FE Design licenses. This functionality is equivalent to the license checkout performed by TOSCA. The FlexLM utilities are only capable to check the exitence and usage of TOSCA licenses. In case all TOSCA licenses are in use, tosca_client does not return until a license can be checked out and given to the license server.

8.3.2.3 Flexlm Test utilities

FlexLM provides a powerful test utility. With the program lmutil (installed in directory <tosca_install_dir>/bin/os_<platform> and the equivalent directory of the TOSCA flexlm installation) the state of the license server and availabilty of the TOSCA licenses can be verified.

For a detailed description of this tool see the flexlm end user manual.

- lmutil lmdiag
  Command:
  lmutil lmdiag [-c <tosca>/license/license.dat]
  In case lmdiag is called without the license path argument, the default path for the current user is used (see lmpath). All license files and license servers in the license

path are queried. lmdiag will prove the installed licenses for availabilty and validity.

- lmutil lmstat
  This gives you additional information about the license server state, with the additional option -a the actual usage of flexlm licenses will be shown.
  In case all TOSCA licenses are in use, you can see which users hold the licenses.
- lmutil lmpath
  If TOSCA can’t find installed licenses, you can prove the license path for the current users with this command.
  lmutil lmpath -status
  This information is saved on unix systems in the HOME directory of the current user in a file .flexlmrc.
  In case the license path is not correct, you can override it with the following option:
  lmutil lmpath -override FED <port@licenseserver>
  This will override the path only for FE Design products.
- lmtools
  Lmtools is a graphical user interface for microsoft windows, which provides comfortable acces to the above mentioned tools .

Version 5.0 Rev. 01 - 03.2004